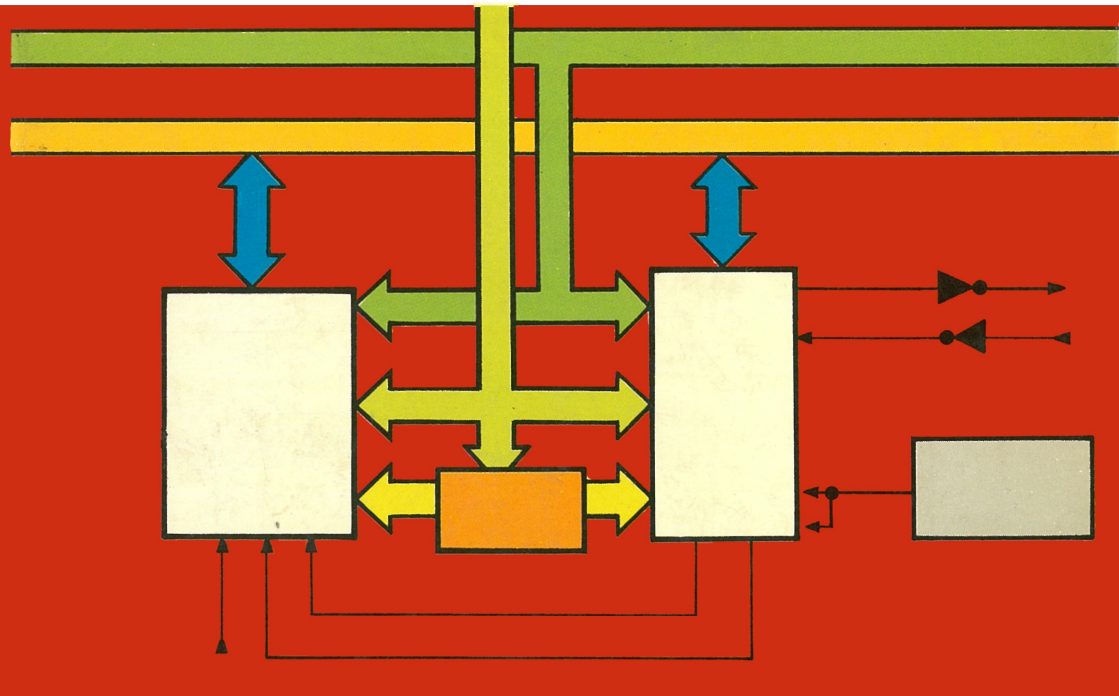
Tecniche d'interfacciamento dei microprocessori

EDIZIONE ITALIANA

AUSTIN LESEA
RODNAY ZAKS

SYBEX

JACKSON ITALIANA EDITRICE



Tecniche d'interfacciamento dei microprocessori

di
Austin Lesea
Rodnay Zaks

JACKSON ITALIANA EDITRICE
Piazzale Massari, 22 - 20125 Milano

RINGRAZIAMENTI

Le persone o ditte di seguito elencate hanno fornito preziose informazioni, fotografie, programmi o schemi dei loro prodotti o progetti. Si ringrazia vivamente per la collaborazione prestata.

Intel, Motorola, Persci (disks), Shugart (disks), Thomson-CSF (schede CRTC), David Reinagel (sintesi musicale), Rockwell, Data I/O (programmatore), Prolog (programmatore), Zilog, Hewlett-Packard (analizzatori), Control Data (disks), North Star, Imsai, Altair (bus S100), Tarbell (interfaccia di cassette), Component Sales (tastiere), MOS Technology, Advanced Micro Devices, Fairchild, NEC, Western Digital, Dynabyte (RAM), National Semiconductor, Analog Devices, Lawrence Laboratory, Università di Berkeley-California, Power One (alimentatori), Fluke, Biomation (analizzatori), Trendar (analisi di guasto).

AVVERTENZE

Si è cercato per quanto possibile, di fornire informazioni complete e rigorose. In ogni caso la Sybex non si assume alcuna responsabilità per il loro impiego; nemmeno al riguardo di infrazioni di brevetti o di altri diritti di terze parti che ne potrebbero derivare. I costruttori di apparecchiature non rilasciano alcuna autorizzazione su apparecchiature protette da brevetto o diritti di brevetto e si riservano la facoltà di cambiare, in qualunque momento, la disposizione circuitale senza alcun preavviso.

In particolare sono soggetti a frequente cambiamento le caratteristiche tecniche e i prezzi. I confronti e le valutazioni sono presentati solo per il loro valore educativo ed i loro principi informativi.

Per le specifiche esatte si rimanda il lettore ai dati del costruttore.

© Copyright per l'edizione originale SYBEX Inc. 1977-1978.
2020 Milvia Street — Berkeley, California 94704

© Copyright per l'edizione italiana SYBEX Inc. 1980.
Tutti i diritti sono riservati — Nessuna parte di questo libro può essere riprodotta, posta in sistemi di archiviazione, trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopiatura etc., senza l'autorizzazione scritta dell'editore.

Stampato in Italia da
S.p.A. Alberto Matarelli — Milano
Stabilimento Grafico

SOMMARIO

PREFAZIONE 5

CAPITOLO 1

INTRODUZIONE 7

Concetti, tecniche in oggetto, Bus; strutture e tecniche operative.

CAPITOLO 2

TECNICHE DI IMPLEMENTAZIONE DELL'UNITÀ DI ELABORAZIONE (CPU) 17

Introduzione, 8080, 6800, Z80, Dynamic Memory, 8085.

CAPITOLO 3

FONDAMENTI DI TRASFERIMENTO DATI SU INTERFACCIA (I/O) 45

Moduli di interfaccia LSI per interfaccia serie e parallela dei sistemi 8080 e 6800, Interruzioni, Controllori di accesso diretto a memoria (DMA), circuiti usati.

CAPITOLO 4

INTERFACCIAMENTO DELLE PERIFERICHE 85

Tastiere, LED, Telescriventi, Lettori di nastro perforato, Motori passo-passo, Registratori a cassette, «Floppy-Disk», Sintetizzatori musicali.

CAPITOLO 5

CIRCUITERIA ANALOGICA — CONVERSIONE ANALOGICA — DIGITALE (A/D) E DIGITALE ANALOGICA (D/A) 239

Introduzione, concetti di conversione D/A, realizzazioni D/A, prodotti reali, conversione A/D, Teorema di campionamento. Approssimazioni successive, integrazione, conversione per confronto diretto, prodotti, Interfaccie per conversione D/A e A/D, Sottosistemi di raccolta dati, risoluzione, offset, conclusioni.

CAPITOLO 6

STANDARD DI INTERFACCIA (BUS) 263

Parallele: S100, 6800, IEEE-488, un esempio di interfaccia di tipo IEEE-488, CAMAC. Seriali: EIA-RS232C, RS422, RS423, Formati sincroni, un esempio di interfaccia di tipo S100.

CAPITOLO 7

STUDIO DI UN CASO: MULTIPLATORE A 32 CANALI 313

Introduzione, specifiche, architettura, Software, Modulo CPU, Modulo RAM, Modulo USART, Modulo di interfaccia di elaboratore, conclusioni.

CAPITOLO 8

ERRATA FUNZIONALITÀ DIGITALE 333

Introduzione, Che tipo di guasto: Componente, rumore, software. Mezzi e metodi: VOM DVM, Oscilloscopio, Probe logici, Analisi mediante indicazioni caratteristiche (Signature analysis), Emulazione, Simulazione, Analizzatore di stati logici, Il perfetto banco di lavoro.

CAPITOLO 9

CONCLUSIONI – EVOLUZIONI 369

I nuovi circuiti integrati: sistemi in singolo modulo, «Plastic Software», la interfaccia programmabile universale.

APPENDICE A 373

Produttori.

APPENDICE B 375

Produttori dell'S100.

APPENDICE C 392

Tabelle di conversione decimale, binaria, esadecimale, ottale.

APPENDICE D 393

Segnali RS232C.

APPENDICE E 394

Segnali IEEE-488.

APPENDICE F 395

Acronimi.

PREFAZIONE

Interfacciare un elaboratore è tradizionalmente un'arte, l'arte di progettare e realizzare l'elettronica di controllo necessaria per connettere periferiche di diverso tipo al modulo centrale di elaborazione.

Con l'avvento dei microprocessori e dei moduli LSI, sin dal 1976, interfacciare i microprocessori non è più un'arte. Significa piuttosto un gruppo di tecniche, e in certi casi solo un gruppo di componenti, da utilizzare nel progetto. Questo libro indica le tecniche e i componenti necessari per assemblare un sistema completo, dalla fondamentale unità centrale di elaborazione ad un sistema equipaggiato con tutte le periferiche comunemente usate, dalla tastiera al «disco - floppy».

La lettura dei capitoli 2 e 3 è particolarmente raccomandata per il progettista che non ha l'esperienza di progetto di un sistema di base. Il secondo capitolo tratta la struttura di una unità centrale di elaborazione (CPU), nel caso di microprocessori così popolari come l'8080, l'8085 ed il Motorola 6800. Il capitolo terzo tratta il gruppo di tecniche di ingresso - uscita usate per comunicare con il mondo esterno, e una breve panoramica dei moduli esistenti che facilitano la realizzazione di tali tecniche.

Il capitolo quarto è fondamentale: la CPU, descritta nella sua implementazione mediante un processore, è interfacciata verso ciascun tipo fondamentale di periferica: tastiera, LED, telescrivente, «disco floppy», schermo raggi catodici, cassetta magnetica.

I capitoli seguenti mettono infine a fuoco specifici problemi e tecniche di interfacciamento, dal progetto industriale (conversione analogico-digitale) (capitolo 5) alla comunicazione con il mondo esterno (tecniche del bus, includendo standard S-100 ed altri standard) (capitolo 6).

Il capitolo settimo sviluppa uno specifico studio di un caso, che comprende i principi di interfacciamento sviluppati nei capitoli precedenti: il progetto di un multiplatore a 32 canali.

Infine, il capitolo ottavo sviluppa le tecniche fondamentali ed i mezzi per rivelare errate funzionalità di sistemi con microprocessori.

Questo libro richiede una conoscenza fondamentale dei sistemi microprocessore, al livello corrispondente a quanto trattato nel libro: **I microprocessori: dai componenti ai sistemi.**

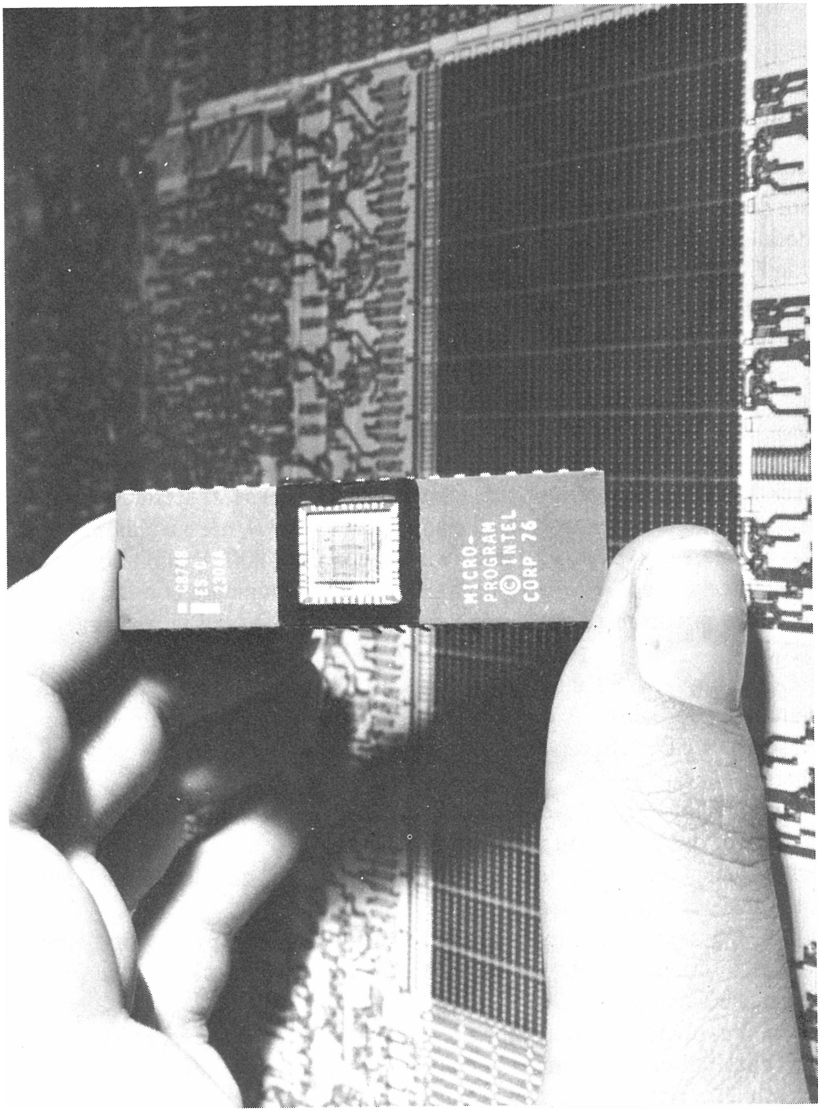


Fig. 1-0: Intel 8748

CAPITOLO 1

INTRODUZIONE

OBIETTIVO

Scopo di questo libro è presentare un gruppo di tecniche necessarie per interfacciare un microprocessore verso il mondo esterno. La disponibilità di nuovi moduli LSI di interfaccia, che realizzano in «hardware» la maggior parte delle tecniche, ha reso semplice la realizzazione dei sistemi.

DA UN'ARTE A UNA TECNICA

Interfacciare microelaboratori è stato tradizionalmente l'arte di realizzare complesse piastre logiche con lo scopo di gestire il trasferimento di dati e di segnali di sincronizzazione, necessari al processore per comunicare con componenti esterni. Il processore ha già esso stesso richiesto tradizionalmente una o più piastre di logica, così come ogni interfaccia di I/O. Tali implementazioni su più piastre sono oggi obsolete in molti casi. L'*integrazione su larga scala* (LSI) comporta oggi lo sviluppo di una CPU integralmente o quasi in un *unico modulo*. Il nuovo mercato creato dal microprocessore ha introdotto, a sua volta, la necessità che i costruttori forniscano i necessari componenti di supporto. Gran parte delle piastre necessarie per realizzare un sistema completo sono ora interfacciate attraverso un modulo LSI. Sin dal 1976 esistono moduli di interfaccia per il controllo di qualsiasi componente. Essi realizzano nell'area delle interfacce quanto il microprocessore ha fatto nel progetto della CPU.

Una piastra di interfaccia completa, o gran parte di essa, è oggi contenuta in alcuni moduli LSI. Il prezzo pagato, proprio come nel caso di un microprocessore, è che l'architettura è congelata entro il modulo LSI.

È adesso possibile realizzare un sistema a microprocessore completo, includendo le interfacce, in un piccolo numero di moduli LSI. *Se sono in fase di sviluppo sistemi con interfacce su una o più piastre logiche, quei progetti sono obsoleti.*

I moduli di interfaccia per microprocessori non hanno ancora raggiunto la maturità. Essi sono ancora moduli «muti». In altre parole possono eseguire un numero molto ridotto di comandi. Si può prevedere che, sulla base di un costo particolarmente basso dell'elemento di elaborazione, gran parte dei moduli di interfaccia verso microprocessori diventerà integralmente programmabile, in un prossimo futuro. Tali moduli prenderanno la sigla «provvisti di microprocessore» e saranno capaci di eseguire sofisticate sequenze programmate. Diventeranno interfacce «intelligenti».

Anche se questo livello non è stato ancora raggiunto, tutte le tecniche presentate in questo volume manterranno la loro validità in futuro. Esisterà sempre una soglia di convenienza tra implementazioni hardware e software. La scelta sarà influenzata dall'introduzione di nuovi componenti e dalle esigenze specifiche delle singole realizzazioni.

IL COMPROMESSO HARDWARE - SOFTWARE

Nel volume saranno indicate in dettaglio le tecniche per risolvere tutti i più comuni problemi di interfaccia. Così come nel progetto di elaboratori, gran parte di tali tecniche possono essere realizzate in *hardware* (usando componenti) oppure in *software* (usando programmi), o ancora con una combinazione dei due criteri. È sempre compito del progettista del sistema trovare un giusto compromesso tra l'efficienza dell'hardware ed il minor numero di componenti in uso in una implementazione software. Saranno sviluppati esempi di entrambe le soluzioni.

CARATTERISTICHE STANDARD DEL SISTEMA MICROPROCESSORE

Nel volume si farà riferimento a un microprocessore standard. Attualmente lo standard consiste in un *microprocessore ad 8 bit*. Esempi sono l'Intel 8080, l'8085, il Zilog Z-80, il Motorola 6800, il Signetics 2650, etc. In base alle limitazioni del numero di terminali del modulo («dual-in line package»: DIP), il microprocessore ad 8 bit è diventato la norma. La ragione è semplice:

Il numero di terminali è limitato a 40 (o 42) per considerazioni economiche. «Tester» industriali necessari per provare componenti con più di 40 terminali non sono infatti disponibili sul mercato, e se lo fossero sarebbero estremamente costosi. Tutti i tester industriali accettano soltanto moduli con al massimo 40 oppure 42 terminazioni.

A causa delle limitazioni di densità realizzabili con processori MOS LSI, non è ancora possibile integrare l'intera memoria necessaria e la circuitazione di I/O entro il modulo microprocessore. Nel sistema standard il microprocessore (MPU), e talvolta il circuito per la temporizzazione (clock), sono gli unici elementi contenuti nel modulo. La memoria di sola lettura (ROM = Read Only Memory) e quella ad accesso casuale (RAM = Random Acces Memory) sono esterni al modulo. Poichè i moduli di memoria e di I/O sono esterni al microprocessore deve essere disponibile un meccanismo di selezione per individuare i componenti: il microprocessore deve disporre di un *bus degli indirizzi*. Lo standard del bus degli indirizzi è di 16 bit, permettendo l'indirizzamento di 64 K posizioni (dove K = 1024. Pertanto $2^{16} = 64$ K).

Un microprocessore ad 8 bit trasferisce 8 bit di dati per volta. Esso sarà pertanto equipaggiato con un *bus dei dati* di 8 bit. Questo richiede altri 8 terminali.

Almeno due terminali sono richiesti per l'alimentazione ed altri due per la connessione ad un oscillatore esterno a cristallo. Infine devono essere presenti $10 \div 12$ circuiti di controllo per la coordinazione del trasferimento dei dati nel sistema (il

bus di controllo). Il numero totale delle terminali ammonta così a 40 senza lasciarne nessuno inutilizzato.

A causa di tale limitazione un microprocessore a 16 bit non può disporre allo stesso tempo di un bus di indirizzi di 16 bit, e di un bus di dati a 16 bit. Uno dei due bus deve essere *multiplato*. Questo comporta un funzionamento più lento e la necessità di componenti esterni per multiplare e demultiplare i bus.

Ci si può aspettare che il progresso nelle tecniche di integrazione porti all'introduzione di un nuovo microprocessore standard, il microelaboratore a *16 bit contenuto in un unico modulo*. Il microelaboratore in un unico modulo è un microprocessore più i circuiti di temporizzazione e le memorie (ROM + RAM) contenuti nello stesso circuito integrato. Poichè la memoria sarà compresa nel modulo non sarà più necessario fornire un bus esterno di indirizzi. 16 terminazioni diventeranno pertanto disponibili per uso diverso. In un tale sistema saranno *disponibili almeno 24 terminazioni per trasferimento dati*. Essi saranno circuiti di I/O per uso generalizzato. L'inconveniente degli attuali microcalcolatori, finchè il problema sarà nei termini indicati, è la limitazione della quantità di memoria che può essere contenuta entro il modulo microcalcolatore. La limitazione attuale è 2048 parole per la ROM e 512 parole per la RAM. Aggiungere memoria esterna comporta una complessa procedura di multiplazione e demultiplazione che non è conveniente per il sistema. Tuttavia, se in un prossimo futuro sarà realizzato un sistema con una memoria significativamente maggiore, ci si potrà aspettare che quello sarà il prossimo standard di progetto.

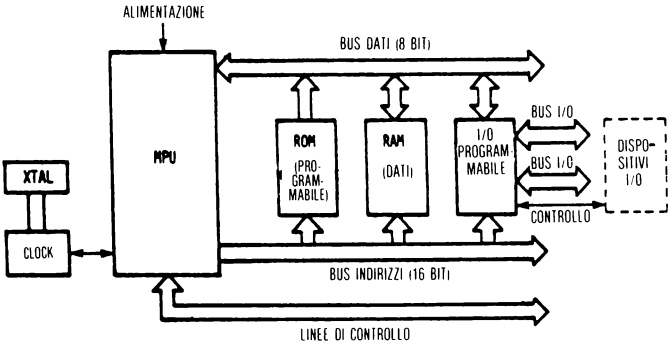


Fig. 1-1: Sistema Microprocessore standard

Nell'ambito della tecnologia attuale il microprocessore ad 8 bit è pertanto lo standard di progetto usato per applicazioni efficienti e flessibili, e ad esso si farà riferimento nel seguito. In figura 1 - 1 è indicato lo schema fondamentale dell'archi-

tettura del sistema standard che sarà utilizzato. Il blocco microprocessore, indicato con MPU è sulla sinistra della figura. In gran parte dei sistemi standard, fino al 1976, il clock era esterno al blocco MPU. In figura esso è indicato sull'estrema sinistra. Dal 1976 il circuito di clock è stato incorporato nel modulo microprocessore stesso e tutti i prodotti recenti non necessitano del clock esterno. Tuttavia essi spesso richiedono un cristallo esterno od un oscillatore. Esso appare in figura connesso al circuito di clock.

Il microprocessore richiede **3 bus.**

Il *bus dei dati* ad 8 bit bidirezionale (implementato in logica tri-state per permettere l'uso di un controllore di memoria ad accesso diretto, o DMAC).

Il *bus degli indirizzi* a 16 bit monodirezionale connesso internamente, entro il microprocessore a puntatori di indirizzo ed, in particolare, al contatore di programma (PC). Il bus degli indirizzi è anch'esso realizzato in una logica tri-state per permettere l'uso del DMAC.

Infine il *bus di controllo* a 10 ÷ 12 linee, che trasferisce i vari segnali di sincronizzazione da e verso il microprocessore. I circuiti di controllo non devono necessariamente essere di tipo tri-state.

Tutti gli usuali componenti del sistema sono connessi direttamente a questi tre bus. I tre componenti fondamentali sono indicati in figura. Essi sono rispettivamente la ROM, la RAM, e il PIO. La ROM è la memoria di sola lettura e memorizza i *programmi.* La RAM è la memoria ad accesso casuale. Essa è una memoria MOS di lettura/scrittura che memorizza i *dati.* Il PIO è un blocco programmabile di I/O che multipla i bus dei dati in due o più porte d'ingresso/uscita. Esso sarà studiato con maggiori dettagli nel capitolo 3. Queste porte possono essere connesse direttamente a componenti d'ingresso/uscita, a controllori di dispositivi o possono richiedere l'uso di circuiti di interfaccia.

I circuiti di interfaccia, o moduli di interfaccia, necessari per interfacciare il suddetto sistema centrale ai dispositivi reali di I/O saranno connessi a questi bus, se i bus del microprocessore, oppure i bus dell'I/O sono realizzati nel PIO, oppure mediante altri moduli.

Le *tecniche di interfaccia* sono a rigore quelle richieste per connettere il sistema centrale, o di base, ai vari apparati di I/O. Le tecniche fondamentali di interfaccia necessarie per connettere un generico sistema a microprocessore ad apparati di I/O sono essenzialmente le stesse. Esse saranno descritte in dettaglio nei capitoli 3, 4, e 5. A livello del singolo microprocessore la logica e l'interfaccia elettrica richieste sono semplici. Tutti i microprocessori standard hanno essenzialmente lo stesso bus di dati e lo stesso bus di indirizzi. Differenze fondamentali esistono solo nel *bus di controllo.* Sono proprio le caratteristiche di tale bus che rendono i blocchi di interfaccia I/O compatibili o meno nei sistemi a microprocessore. Come esempio di caratteristiche fondamentali di interfaccia in figura 1-2 sono rappresentate quelle dell'8080, del 6800, e ancora del 6502 SC/MP.

I dispositivi di interfaccia I/O richiedono la conoscenza di due tecniche fondamentali:

1. La struttura di una CPU completa, comprendente un modulo microprocessore. Questo argomento sarà trattato nel capitolo 2.
2. Le tecniche fondamentali di I/O usate per comunicare tra il microprocessore ed il mondo esterno. Questo argomento sarà trattato nel capitolo 3.

SEGNALI DI CONTROLLO DEL MICROPROCESSORE

È stato indicato che una MPU standard crea tre bus: il bus dei dati ad 8 bit bidirezionale, il bus degli indirizzi a 16 bit monodirezionale e il bus di controllo di dimensione variabile, dipendente dal tipo di microprocessore. Il bus dei dati è essenzialmente identico per tutti i microprocessori. Esso è un bus bidirezionale ad 8 bit, normalmente implementato in logica a tre livelli. Similmente, il bus degli indirizzi è quasi universalmente a 16, talvolta 15, bit monodirezionale, usato per selezionare dispositivi esterni alla MPU. Le modalità d'uso e le relazioni tra il bus di indirizzi e il bus di dati saranno indicate nel prossimo capitolo. Il terzo bus è l'unico che presenta una certa complessità. Esso trasferisce segnali di controllo o segnali di interfaccia.

Il bus di controllo svolge 4 funzioni:
1. sincronizzazione di memoria
2. sincronizzazione di I/O
3. schedulazione dell'attività della MPU - interruzione e DMA
4. funzioni di utilità, quali il clock e il reset.

La sincronizzazione della memoria e dell'I/O sono essenzialmente simili.

È usata una procedura di «hand - shake». In un'operazione di lettura, un segnale o stato di disponibilità (ready) indicherà la disponibilità di dati. I dati saranno pertanto trasferiti sul bus dei dati. Per alcuni dispositivi di I/O è generata una conferma (acknowledge), per confermare la ricezione dei dati. Per operazioni di scrittura, la disponibilità del dispositivo esterno è verificata attraverso un bit di stato o un segnale, e il dato è successivamente depositato nel bus dei dati. In tal caso può essere generata anche una conferma dal dispositivo per indicare l'avvenuta ricezione dei dati.

La generazione, o la sua assenza, di una conferma è tipica di una procedura sincrona rispetto ad una asincrona. In una procedura sincrona tutti gli eventi hanno luogo entro un preciso intervallo di tempo. In tal caso non è necessaria una conferma. In un sistema asincrono deve invece essere generata una conferma. La scelta tra una filosofia di comunicazione sincrona o asincrona è fondamentale per il progetto del bus di controllo. Il progetto sincrono ha la potenzialità di una velocità maggiore e di un minor numero di circuiti di controllo. Tuttavia esso impone limitazioni di velocità ai dispositivi esterni. Il progetto asincrono richiede conferme addizionali di trasferimento, e talvolta più logica, ma permette la interconnessione entro lo stesso sistema di componenti a velocità diversa.

EQUIVALENZE APPROSSIMATE DEI SEGNALI

	8080&8228	8085	Z-80	6800	6502
INDIRIZZO	A0–A15	AD0-AD7 +ALE A8-A15	A0-A15	A0-A15	AB0-AB15
DATI	D0-D7	AD0-AD7 +–ALE	D0-D7	D0-D7	DB0-DB7
CONTROLLO	HLDA	HLDA	BUSAK	BA&VMA	----------
	HOLD	HOLD	BUSRQ	HALT	RDY
	φ2	CLK	––	φ2 stirato	φ2 stirato
	INT	INTR	INT	IRQ	IRQ
	INTE	––	––	––	––
	WAIT	––	––	––	––
	READY	READY	WAIT	––	RDY
	RESET	RESET	RESET	RESET	RESET
	SYNC	––	M1	––	SYNC
	INTA	INTA	M1&IORQ	VMA&FFF8	––
	MEMR	RD&IO/M	RD&MEMRQ	R/W&φ2	R/W&φ2
	MEMW	WR&IO/M	WR&MEMRQ	come sopra	come sopra
	I/O RD	RD&IO/M-	RD&IORQ	come sopra	come sopra
	I/O WR	WR&IO/M-	WR&IORQ	come sopra	come sopra
	BUSEN	––	––	HALT	––
	SSTB	––	––	––	––
ALTRI SEGNALI DI CONTROLLO	––	RST 5.5	––	––	––
	––	RST6.5	––	––	––
	––	RST 7.5	––	––	––
	––	TRAP	NMI	NMI	NMI
	––	RESET	––	––	––
		OUT			
	––	SID	––	––	––
	––	SOD	––	––	––
	––	ALE	––	––	––
	––	––	RFSH	––	––
	––	––	HALT	––	––
	––	––	––	TSC	––
	––	––	––	DBE	––
	––	––	––	––	SO

Fig. 1-2: Equivalenze dei segnali

Un esempio di segnali di controllo è indicato in figura 1-3, relativamente all'8080.

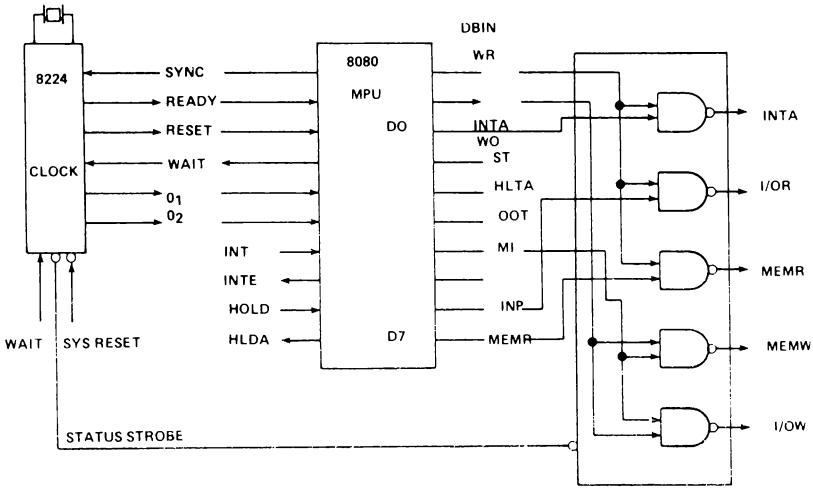


Fig. 1-3: Segnali di controllo dell'8080

In figura 1-4 e 1-5 sono indicate le temporizzazioni necessarie. In figura 1-6 e 1-7 sono invece indicati i segnali del bus del 6800. Le caratteristiche di tali bus saranno descritte nel secondo capitolo. Il capitolo sesto fornisce ulteriori informazioni sui bus e descrive alcuni standard relativi attualmente usati.

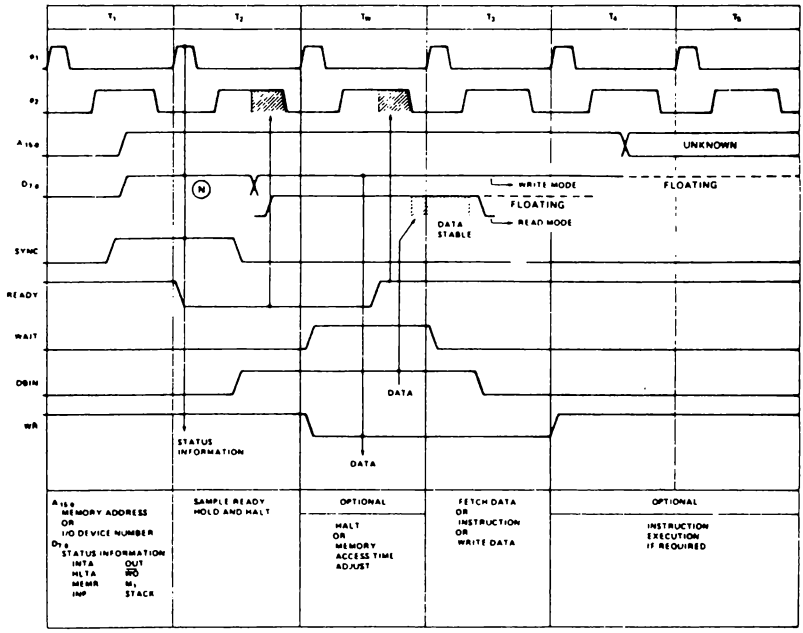


Fig. 1-4: Ciclo di istruzione base dell'8080

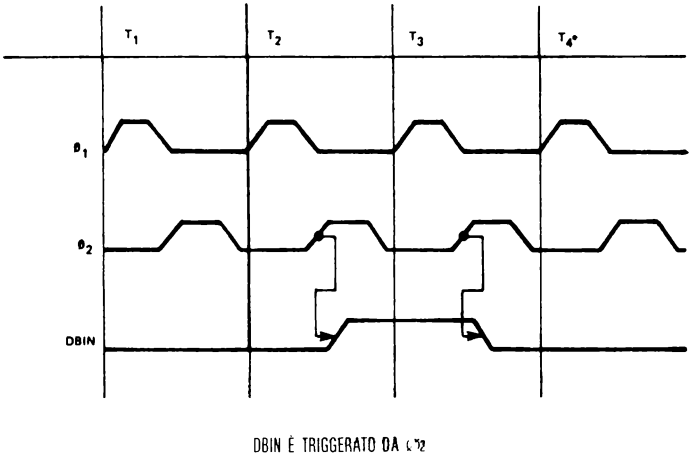


Fig. 1-5: Temporizzazione di DBIN

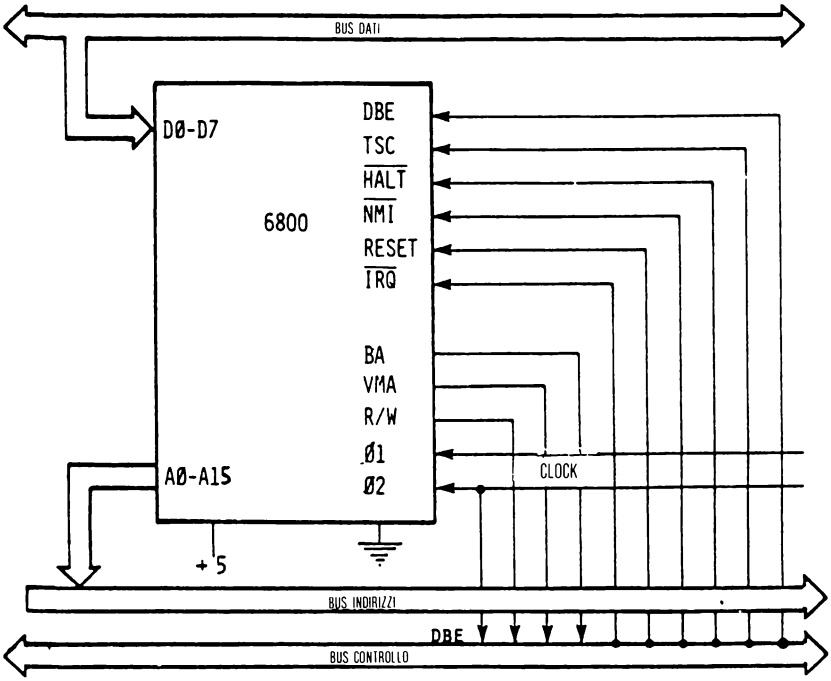


Fig. 1-6: Segnali del bus del 6800

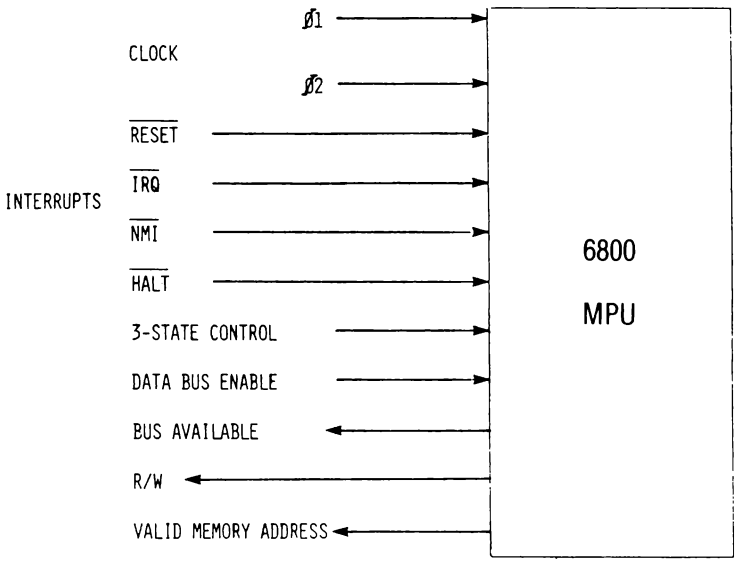


Fig. 1-7: Segnali del bus di controllo del 6800

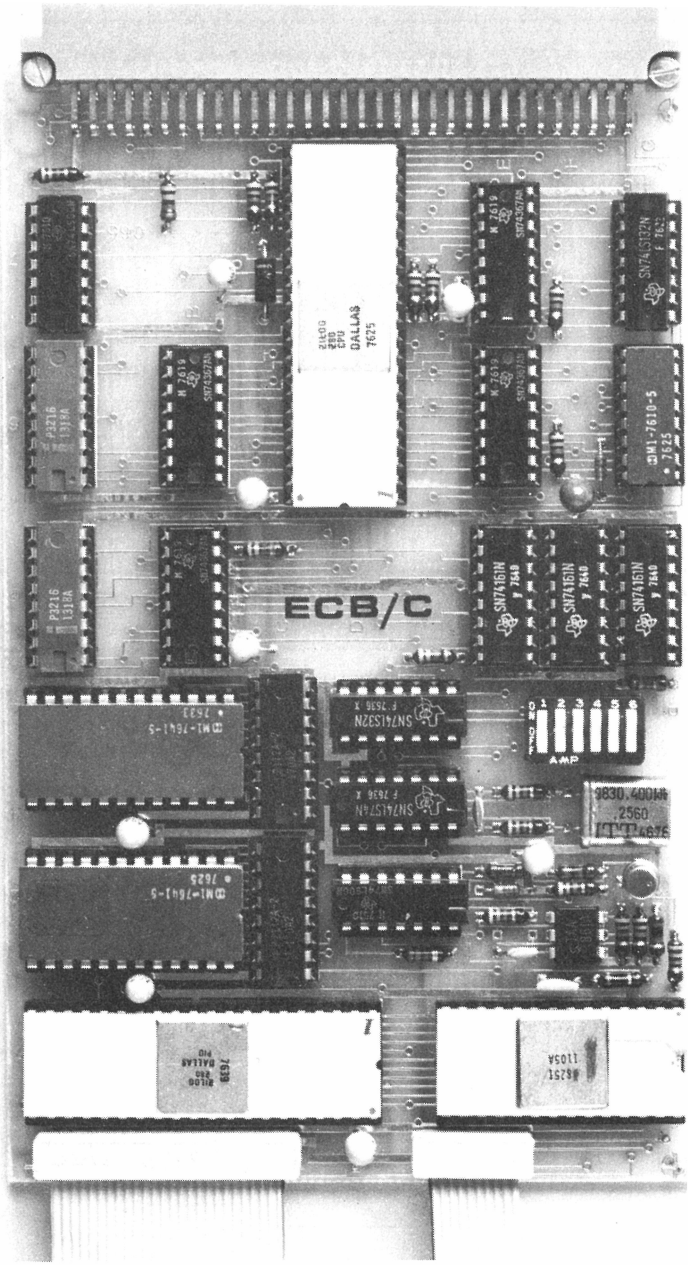


Fig. 2-0: Piastra CPU dello Z-80

CAPITOLO 2

STRUTTURE E CARATTERISTICHE DELLE UNITÀ CENTRALI DI ELABORAZIONE (CPU)

INTRODUZIONE

Il nucleo dei sistemi con microprocessore è la *unità centrale di elaborazione* (Central Processing Unit - CPU). La CPU comprende il microprocessore, oltre ad altri componenti addizionali che il sistema richiede. Dispositivi di memorizzazione, memorie tampone (buffers), decodificatori, generatori di clock, sono tutti presenti in una tipica unità centrale. Gran parte di tali circuiti sono ora integrati in un unico modulo insieme al microprocessore. Infatti, sin dal 1976 sono una realtà microcalcolatori in un unico modulo. Tuttavia, pur con l'avvento del microcalcolatore in un unico modulo, esistono alcune limitazioni nella fabbricazione di circuiti integrati. Esistono, infatti, tre limiti fondamentali nell'attuale tecnologia LSI. Il *livello di integrazione* limita il numero dei transistori per modulo, il *contenitore* limita il numero delle terminazioni per modulo, e il tipo di *materiale del substrato* esclude la integrazione di alcuni dispositivi.

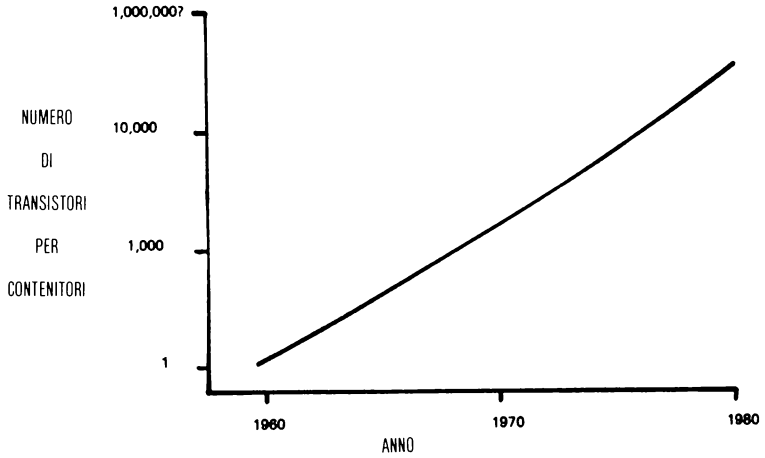


Fig. 2-1: Evoluzione della integrazione di componenti

Inizialmente furono integrati in un unico modulo soltanto singoli transistori. Successivamente entrarono nel mercato coppie differenziali e semplici porte logiche. La tecnologia attuale permette invece che fino a 30.000 dispositivi siano integrati in un singolo modulo. La figura 2-1 rappresenta il numero di componenti integrati per modulo nell'arco di 20 anni. Un unico fattore è rimasto costante in questo processo: i difetti di processo limitano la massima dimensione del singolo modulo. Le rese sono maggiori quanto minore è la dimensione del modulo. Nel progetto di un modulo LSI la dimensione reale (dimensione del circuito integrato) gioca un ruolo molto importante che influenza il costo finale del modulo. La figura 2-2 indica la relazione tra la resa e la dimensione del modulo. La resa cresce anche con l'esperienza di produzione indicata con il termine «curva di apprendimento». Il costo diminuisce al crescere delle quantità, per l'accresciuta resa.

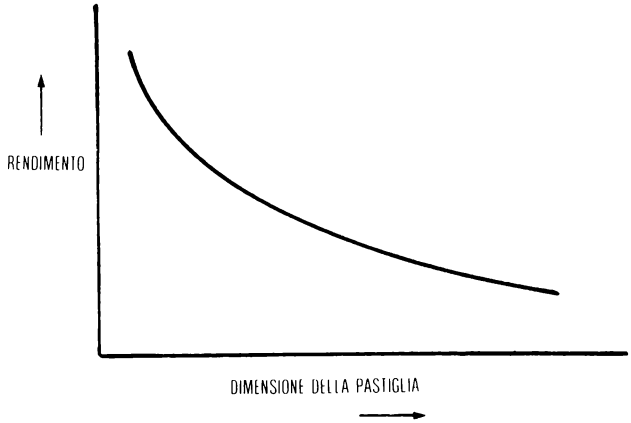


Fig. 2-2: Resa rispetto alla dimensione del modulo

Un fattore meno ovvio è *l'incapsulamento* del dispositivo LSI. Gli attuali apparati di test non possono gestire moduli con più di 40 terminazioni. Sistemi di test futuri potranno superare tale limitazione, ma attualmente la limitazione di terminazioni per modulo richiede l'uso di tecniche di multiplazione: il bus dei dati deve essere usato anche per trasferire *indirizzi* o *controlli,* per le limitazioni delle vie disponibili (esempio: 8080, 8085).

In che modo influenza la tecnologia LSI il materiale costituente il substrato? Alcuni componenti richiedono un materiale differente per il substrato. L'esempio più semplice è offerto dal cristallo richiesto dalla temporizzazione. Un cristallo è ottenuto mediante taglio di un quarzo, mentre il circuito integrato è realizzato dal silicio. Pertanto un sistema che richiede accurata temporizzazione necessita di un cri-

stallo che a causa dell'incompatibilità è esterno. Oltre alla suddivisione del nostro sistema in più componenti a causa della limitazione della tecnologia LSI, spesso sono richiesti altri componenti addizionali per l'espansione del sistema. Complessi sistemi a microprocessori richiedono infatti un significativo ammontare di logica di supporto.

Questo capitolo indica i concetti, le tecniche, ed i componenti necessari per realizzare una CPU completa: dall'architettura di sistema al supporto logico. Saranno considerati 4 sistemi tipici, usanti rispettivamente l'8080, il 6800, lo Z80 e l'8085.

ARCHITETTURA DI SISTEMA

La figura 2-3 indica lo schema blocchi di un tipico sistema a microprocessori. Tutti quelli standard, come l'8080 o il 6800 hanno una architettura simile. Tre bus connettono i componenti del sistema: dati, indirizzi e controlli.

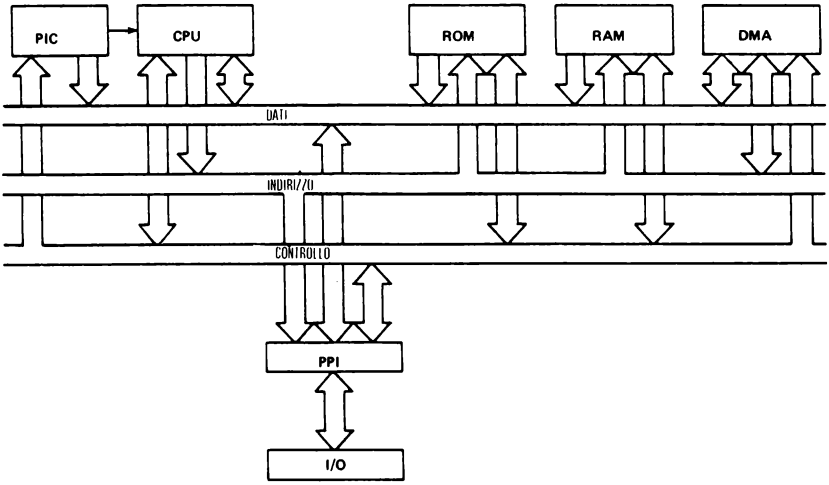


Fig. 2-3: Architettura di un sistema tipico

Il *bus dei dati* trasferisce informazioni da e verso l'elemento processore. Esso trasferisce le istruzioni prelevate da memoria, i dati di ingresso dai moduli periferici, i dati memorizzati entro la memoria, e i dati in uscita da fornire ai moduli periferici.

Per specificare dove i dati devono essere consegnati, o da dove essi sono prelevati, è usato il *bus di indirizzo*. Esso seleziona una posizione in memoria o un registro

o ancora un modulo di ingresso/uscita. Il bus di controllo è usato per controllare la sequenza e il tipo di operazione che deve essere eseguita. Il bus di controllo indica in particolare il tipo di operazione da realizzare: «leggi da memoria e trasferisci al processore», «scrivi in memoria prelevando dal processore», «leggi da un modulo di ingresso e trasferisci al processore», o «scrivi su un modulo di uscita prelevando dal processore». Inoltre, interruzioni, accessi diretti a memoria, e altre funzioni di controllo sono trasferite attraverso circuiti del bus di controllo, per realizzare la schedulazione e la sincronizzazione degli eventi.

Il nostro microprocessore ha 8 vie per dati, 16 vie per indirizzi e almeno 8 vie di controllo. Otto bit di dati formano un *byte*. Il byte è l'unità di informazione nel nostro sistema standard. Metà di un byte è talvolta conosciuto come *«nibble»*. Le sedici vie di indirizzo permettono l'indirizzamento di 65.536 (2^{16}) differenti posizioni di memoria o bytes. Due metodi sono usati per selezionare una posizione di memoria, o un registro del modulo: selezione lineare, o una selezione integralmente decodificata.

Selezione lineare

Nel mondo dei microprocessori, la memoria è ripartita in memoria a sola lettura (ROM) per programmi e tabelle di dati fissi, e in memoria ad accesso casuale (RAM) per memorizzazione di dati o informazioni temporanee, a causa della volatilità delle MOS RAM.

Quando è usato più di un tipo di memoria, i due tipi di memoria sono normalmente in moduli separati. Inoltre, la dimensione di ciascuna di esse è considerevolmente minore delle 65.536 possibili posizioni del nostro sistema. Occorre allocare ciascun componente in una sua propria posizione nella nostra *mappa di memoria*. La mappa di memoria è il metodo di indirizzamento per i bit del bus di indirizzi.

Inizialmente, ciascun componente, RAM o ROM, avrà 256 posizioni. Ciò implica che otto vie di indirizzo saranno necessarie per selezionare una delle possibili 256 posizioni in ciascun modulo. Oltre che queste 8 vie il processore deve poter selezionare uno specifico componente. I componenti ROM o RAM hanno, oltre che i propri indirizzi di ingresso, almeno anche un «selettore di modulo» (C.S.). Questa via di selezione, quando attivata, permette che sia realizzata l'operazione sul componente (lettura o scrittura).

Sono usate due tecniche fondamentali per selezionare il modulo: la *selezione lineare* connette singole vie di indirizzo a specifici ingressi di selezione di modulo. Per esempio se è connesso ad un selettore di modulo il bit di indirizzo più significativo (il bit 15), quel modulo è selezionato ogni qualvolta il bit più significativo è a livello uno. Questo accade per metà delle complessive posizioni di memoria.

Assumendo che la ROM sia selezionata quando il già indicato bit di indirizzo più significativo è a zero e la RAM quando è a livello uno, l'indirizzamento delle 256 posizioni disponibili entro ciascun componente è realizzato connettendo le vie da A0 ad A7 al bus di indirizzo.

Il vantaggio essenziale della selezione lineare è la semplicità: non è necessaria logica supplementare per selezionare il modulo.

Ciascun nuovo modulo è selezionato mediante una via di indirizzo dedicata. Questo approccio si presta per piccoli sistemi con microprocessore.

Per esempio, supponiamo che siano utilizzati un modulo ROM 1K x 8, una RAM 512 x 8 e due moduli di periferia. La ROM richiede 10 vie per la selezione degli indirizzi A0-A9, più una via per la selezione del modulo: A 14. La RAM userà A0-A8 per la selezione dell'indirizzo e A15 per la selezione del modulo. Le vie A12, A13, A14, A15 saranno usate per componenti addizionali.

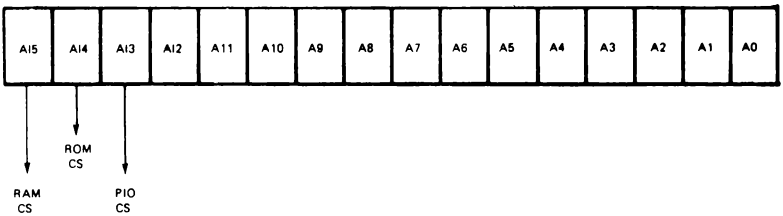


Fig. 2-4: Selezione lineare

Purtroppo la selezione lineare divide la memoria disponibile a metà ogni volta che è usata una specifica via di indirizzo. Se è necessario selezionare più componenti di quanto sia possibile mediante le vie di indirizzo, deve essere usato un altro metodo: «*l'indirizzamento a decodifica integrale*».

Indirizzamento a decodifica integrale

Lo scopo dell'indirizzamento a decodifica integrale è quello di poter indirizzare integralmente 64 K posizioni.

Nel nostro esempio, la RAM a 256 posizioni risiede nelle ultime 256 posizioni della memoria. Espresso in binario il campo degli indirizzi è da 1111111100000000_2 a 1111111111111111_2. Raggruppando in gruppi di bit e convertendo in esadecimale diventa: da FFOO a FFFF. (Vedere l'appendice per la tabella di conversione in esadecimale). Noi vediamo che la RAM può essere abilitata quando tutti gli 8 bit di ordine superiore sono eguali ad «1». Realizzando l'AND questi bit nel loro insieme realizzeranno la selezione del modulo. La figura 2-5 illustra la decodifica per il nostro esempio.

Invece di usare porte AND per ogni componente, esistono componenti di uso generale che realizzano funzioni di porta, cioè i «*decodificatori*». Un esempio è costituito dall'8205 o dal 74LS138, decodificatori 3/8.

L'8205 ha tre ingressi per selezionare una tra otto uscite mutualmente esclusive, in funzione di tre ingressi di abilitazione. Quando i tre ingressi di abilitazione sono

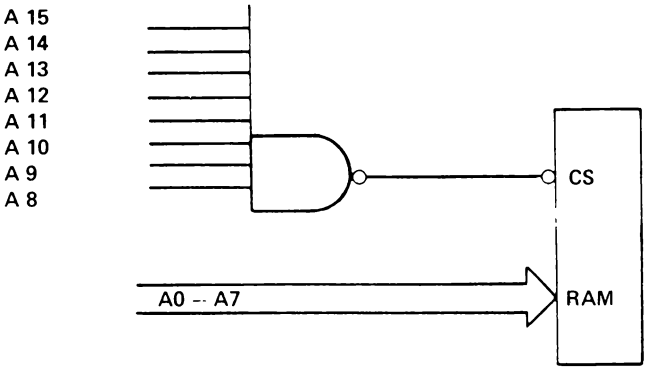
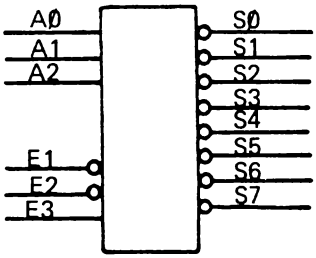


Fig. 2-5: Selezione integralmente decodificata

nei loro stati opportuni, una delle uscite sarà attiva in relazione alla configurazione delle tre vie di selezione. Saranno presentati esempi di uso dell'8205 nella sezione hardware del volume, per chiarire gli schemi a decodifica completa.

Mediante la decodifica completa i componenti sono selezionati senza eccessiva riduzione dello spazio di indirizzi disponibile. Possono essere organizzate aree di



$$S\emptyset = (\overline{A\emptyset} \cdot \overline{A1} \cdot \overline{A2}) \cdot (\overline{E1} \cdot \overline{E2} \cdot E3)$$

$$S1 = (A\emptyset\ \overline{A1} \cdot \overline{A2}) \cdot (\overline{E1} \cdot \overline{E2} \cdot E3)$$

$$\vdots$$

$$S7 = (A\emptyset \cdot A1 \cdot A2) \cdot (\overline{E1} \cdot \overline{E2} \cdot E3)$$

Fig. 2-6: Decodificatore 8205

memoria contigue nelle quali l'indirizzo è commutato da un modulo al successivo senza aree di memoria non esistenti o ricoprenti uno spazio parzialmente sovrapposto («overlapping»). Gran parte dei sistemi sono implementati con una tecnica mista di selezione lineare e di decodifica parziale.

Moduli di memoria

I componenti fondamentali attualmente usati per la memorizzazione delle informazioni sono la RAM e la ROM. La ROM contiene informazioni permanenti che *non possono essere cambiate* dal sistema. La RAM permette la memorizzazione e l'utilizzazione di informazioni memorizzate. Il programma del sistema è normalmente mantenuto in una ROM, non volatile, finchè non devono essere apportati cambiamenti. I dati ed i risultati intermedi sono memorizzati in RAM.

Il termine «RAM» è normalmente associato a componenti a semiconduttore, ma è anche usato per altri supporti di memorizzazione come ad esempio la memoria o nuclei.

Un modulo RAM può contenere da 256 a 16384 celle, e ciascuna di esse rappresenta un bit della parola di informazione o byte. Ciascuna cella può essere composta da un «flip-flop», e in tal caso è un *componente statico,* oppure da una struttura di tipo capacitivo e in tal caso essa è un *componente dinamico.* La RAM statica mantiene l'informazione finchè è presente l'alimentazione, mentre un componente dinamico richiede che sia ripristinata la carica memorizzata in ciascuna cella. Tale procedura è chiamata «refreshing». Ciò significa che la memoria dinamica richiede un ciclo di «refreshing» dall'uno al cinque per cento del tempo.

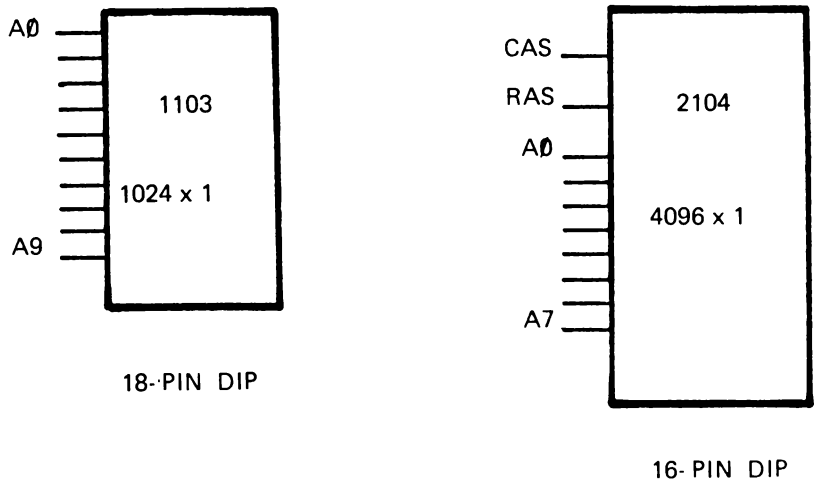


Fig. 2-7: RAM dinamiche della Intel

Questo è importante in alcune applicazioni in tempo reale, quando la memoria è impegnata e non disponibile, mentre si sta eseguendo il ciclo di ripristino. La ROM costituisce un modulo LSI, ma può anche essere usata per indicare altri tipi di memorie a solo lettura. Sono infatti disponibili diversi tipi di ROM. La ROM realizzata mediante maschere di codifica è programmata dal *costruttore* e resta programmata per l'intera vita del componente. Essa non può essere alterata. La PROM è programmata dall'utente e questo può essere realizzato con una tecnica simile a quella della bruciatura di un fusibile. Un bit è infatti programmato facendo saltare un fusibile microscopico. Altra tecnica è quella ad accumulo di carica. Essa trattiene la trama di codifica per decine di anni. Quest'ultima tecnica è anche conosciuta come EPROM perchè può essere cancellata la trama con raggi ultravioletti e ricodificata. La EPROM è cancellabile elettricamente e può essere considerata come una RAM, eccetto che sono necessari tipicamente 100 millisecondi o più per cancellare la codifica. Questo non la rende conveniente per l'uso come memoria tampone («scratchpad») per calcoli o manipolazione di dati. L'uso dell'EPROM è attualmente ristretto ad applicazioni militari.

Bufferizzazione dei bus

Ciascun ingresso di un componente costituisce un carico per l'uscita che lo pilota. Gran parte dei componenti pilotano da uno a 20 altri componenti. Ogni componente deve essere scelto in base ad una verifica delle sue capacità di pilotaggio ed alla sua impedenza di uscita.

I bus dei microprocessori devono usualmente essere connessi a tutti i moduli di memoria e di ingresso-uscita di periferiche presenti nel sistema. Tutti i microproces-

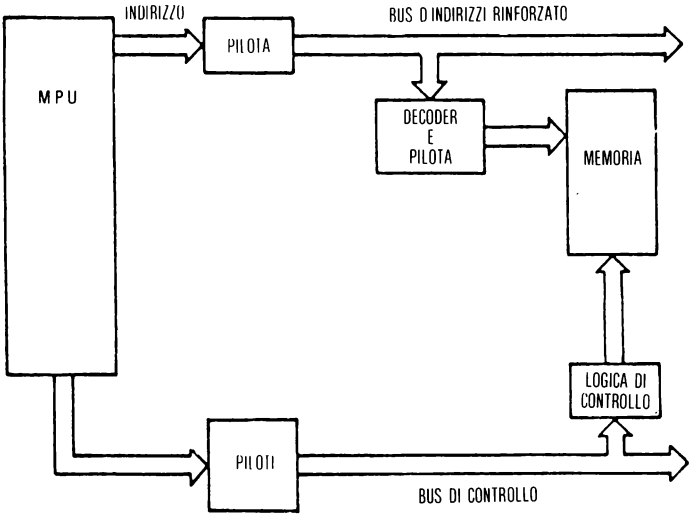


Fig. 2-8: Rigenerazione degli indirizzi e vie di controllo

sori non offrono l'intera capacità di pilotaggio in uscita necessaria per un grosso sistema. Pertanto sono usate *memorie temporanee* («buffer») o *«circuiti di pilotaggio»* per trasformare la capacità di pilotaggio dei bus.

Esistono *trasmettitori* di bus per sopperire alla limitata potenza del bus, e *ricevitori* di bus per ricevere dal bus e pilotare circuiti del processore.

La figura 2-8 illustra l'uso di trasmettitori per memorizzare temporaneamente le informazioni dei bus di indirizzo e di controllo. Le vie degli indirizzi e dei controlli sono *unidirezionali:* i dati fluiscono in un'unica direzione.

La figura 2-9 illustra l'uso di ricetrasmettitori per il bus dei dati. I dati devono infatti passare in entrambe le direzioni e pertanto sono usati sia trasmettitori che ricevitori. Il bus *bidirezionale* dei dati riceverà e trasmetterà dati, in relazione alla funzione che deve essere realizzata.

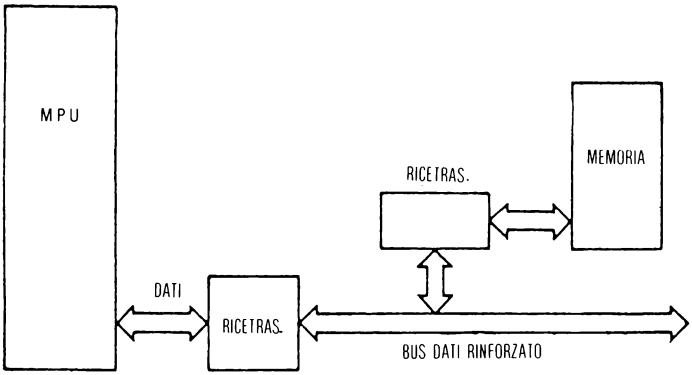


Fig. 2-9: Rigenerazione del bus dati

Il concetto di una architettura di sistema sarà espanso e completato nel capitolo tre che tratta le tecniche di ingresso e uscita. Per chiarire i concetti fino ad ora presentati, sono ora descritte le strutture di quattro sistemi reali: l'8080, il 6800 e lo Z80 (con l'uso di RAM dinamica), e un sistema 8085.

IL SISTEMA 8080

Il sistema 8080 è stato lo standard di architettura di microprocessore più ampiamente usato. L'8080 è un processore molto conosciuto, usato anche in molti microcalcolatori per hobby. Noi tratteremo la struttura dell'intero modulo di elaborazione centrale per un tipico sistema di elaborazione 8080. Saranno indicate le modalità di connessione del clock, del controllore di sistema, la RAM e la ROM.

L'ingresso e l'uscita saranno trattate in dettaglio nel capitolo 3.

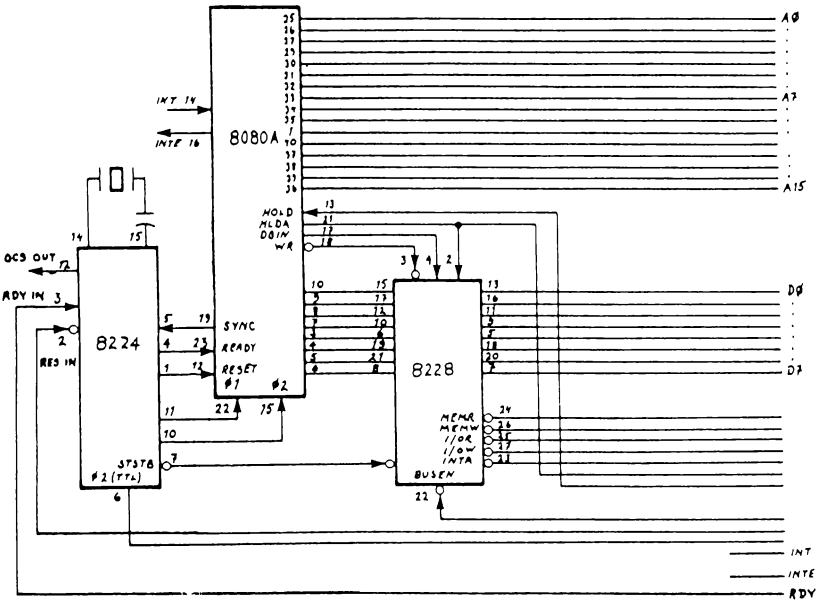


Fig. 2-10: CPU completa con 8080

Il clock

L'8080 richiede un clock a due fasi non sovrapposte. Il clock deve oscillare tra +11 e +0,3 volt. Il clock è pertanto incompatibile con la logica TTL. Inizialmente i generatori di temporizzazione furono implementati con componenti discreti o con speciali circuiti di pilotaggio integrati.

La INTEL introdusse il circuito di orologio 8224 per ridurre l'hardware e per semplificare l'interfaccia di orologio. Occorre semplicemente connettere il cristallo all'8224, l'8224 all'8080 e l'intera interfaccia temporizzatore è completa. La connessione dell'8224 è indicata in figura 2-10 e l'intera struttura dell'8224 è indicata in figura 2-11.

Il modulo di controllo del sistema

Nel progetto dell'8080 la mancanza di terminazioni è stata la maggiore limitazione. Per abilitare l'uscita dei circuiti di controllo necessari, è stato necessario multiplare le uscite sulle terminazioni.

Le funzioni di controllo e di indirizzo hanno dovuto condividere le vie del bus dei dati.

In tal caso il progettista ha scelto di multiplare informazioni di controllo e lo *stato* sul bus dei dati. Questo byte di stato deve essere controllato per l'abilitazione al-

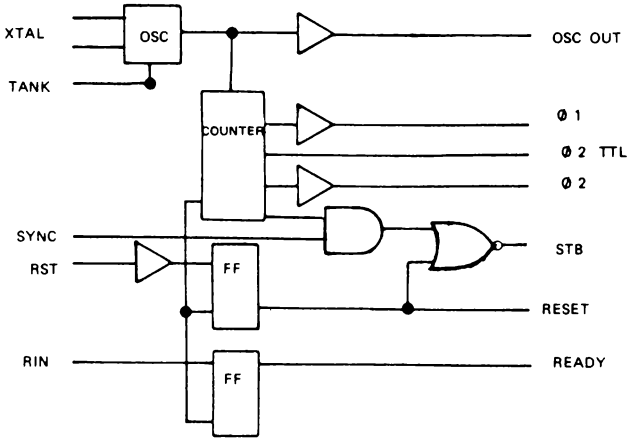


Fig. 2-11: Schema dell'8224

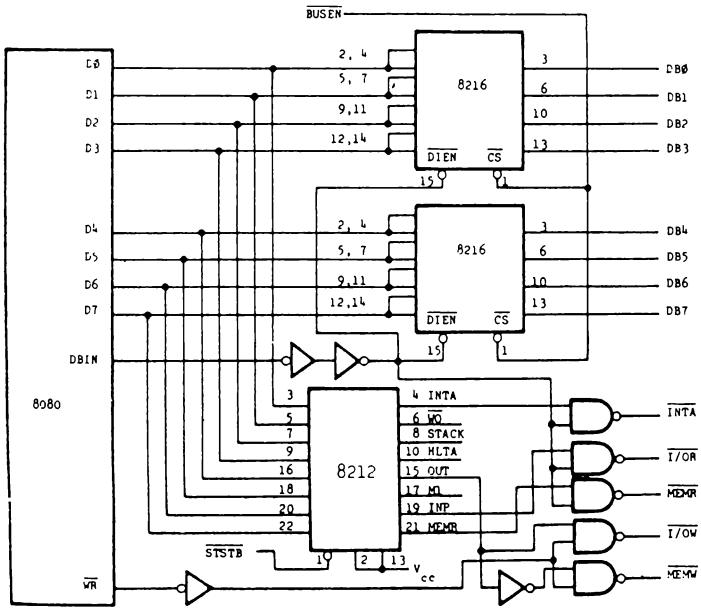


Fig. 2-12: Controllore di sistema usando l'8212 e 8216

l'uso del bus dei dati quando è disponibile il segnale SYNC. La mancanza di terminazioni è essenzialmente dovuta alla tecnologia di base usata nell'8080, che richiede tre livelli di potenza, usando pertanto 8 terminazioni.

I primi progetti di microprocessori usavano, per rivelare tali segnali di stato, logiche di trasferimento controllato e logica random. La compatibilità con tali tecniche hanno suggerito che il progetto del bus S100 mantenesse quanto è stato indicato *vecchi segnali di stato* dell'8080. In figura 2-12 è indicato lo schema di quello che è conosciuto come *modulo di controllo del sistema*. Il controllore di trasferimento memorizza le informazioni di stato e le porte decodificano lo stato sulla base anche di altre linee di controllo dell'8080 e producono segnali di controllo per i moduli di memoria e di ingresso - uscita.

La Intel, ritenendo inizialmente che la funzione di controllo del sistema sarebbe stata integrata in un unico circuito integrato, ha introdotto il modulo 8228, indicato in figura 2-13. Questo componente controlla il trasferimento dello stato e pilota il bus di controllo. Esso, inoltre, contiene una memoria tampone per il bus dei dati, cioè svolge anche le funzioni di pilotaggio del bus dei dati.

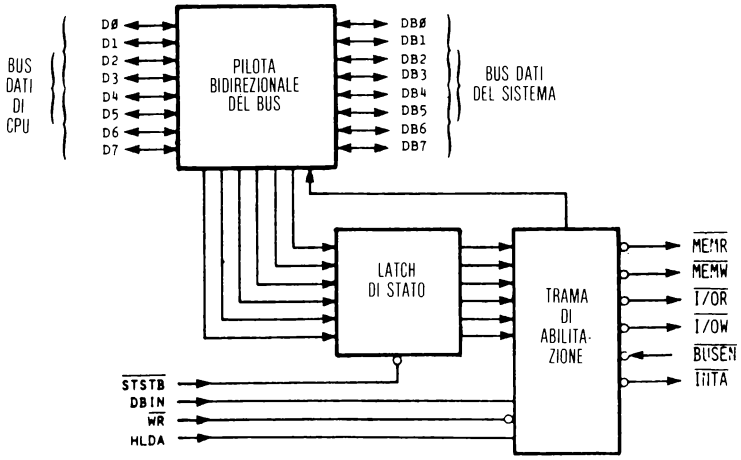


Fig. 2-13: Controllore di sistema 8228

Il trio composto dall'8224, dall'8228 e dall'8080 assolvono interamente le funzioni del processore centrale. L'unico componente richiesto è il cristallo dell'oscillatore. Per completare la CPU è necessario aggiungere la memoria di programma e la memoria ad accesso casuale (ROM oppure RAM).

Connessione della ROM

Le memorie a sola lettura si presentano in due tipi fondamentali: programmabili o facenti uso della tecnica di mascheratura. Le ROM programmabili possono essere programmate una sola volta quando devono essere usate (tale è il caso delle ROM con fusibile o delle PROM); oppure possono essere programmate, usate e

cancellate (come le ROM cancellabili con raggi ultravioletti o le EPROM).

Le ROM con mascheratura sono programmate in fase di produzione di sistemi dal costruttore.

Le ROM cancellabili o con fusibili sono usate per i prototipi.

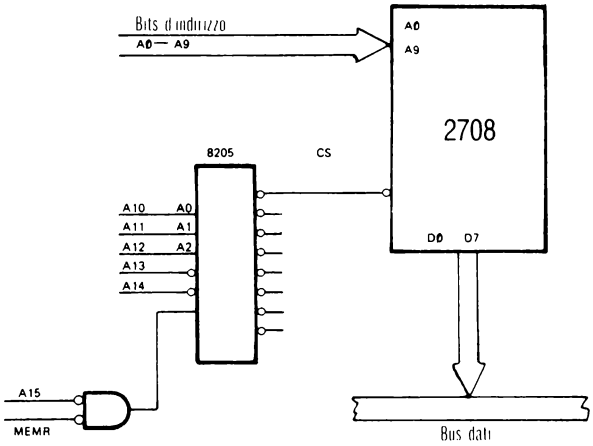


Fig. 2-14: Selezione del 2708 usando l'8205

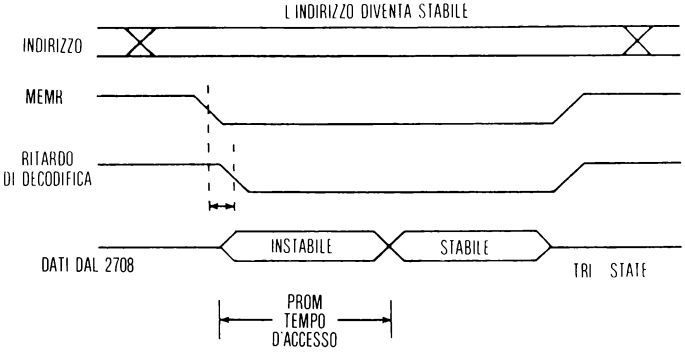


Fig. 2-15: Temporizzazione della PROM

In figura 2-14 è indicata la modalità di connessione al nostro 8080 di una ROM cancellabile tipica. Questo componente, una EPROM 2708; contiene una memoria di 1024 bytes. Per indirizzare 1024 bytes sono necessarie 10 vie di indirizzo (2^{10}=1024). Inoltre il circuito integrato deve essere selezionato nella sua esatta posizione entro la mappa di memoria.

Noi, per esempio, sceglieremo di porre la memoria nelle posizioni tra 0000 ed 03FF in esadecimale. Per decodificare questo spazio di indirizzi è usato un 8225 oltre ad una logica di selezione per controllare le fasi di lettura della memoria. È da notare che esso può selezionare fino a sette moduli 2708 addizionali, se necessari. Gli indirizzi saranno adiacenti. Il bus dei dati è connesso direttamente alle vie dei dati del modulo di controllo del sistema 8228. L'unica via di controllo necessaria è quella di lettura di memoria. La temporizzazione della lettura della memoria è indicata in figura 2-15.

Le vie di indirizzo e di abilitazione-lettura rendono attivo il 2708. Dopo un lasso di tempo chiamato tempo di accesso, il byte di dati prelevati saranno presenti nel bus dei dati. Il processore legge tale byte ed esegue l'istruzione.

Connessione della RAM

La dimensione conveniente per una economica produzione di una ROM è di 1K di 8 bit (1K=1024). Le RAM, invece, presentano diverse dimensioni. La configurazione a minimo costo è quella di un K a singolo bit (minimo numero di terminazioni). Poiché un byte è composto da 8 bit sono necessari 8 componenti, uno per ciascun bit. Altra dimensione comune è quella di 256 per 4 bit. Questo tipo di RAM è quello qui trattato. 256 per 4 implica che sono necessari due componenti per completare il byte. In figura 2-17 è indicato lo schema di connessione al bus dell'8080 di memorie 256 per quattro.

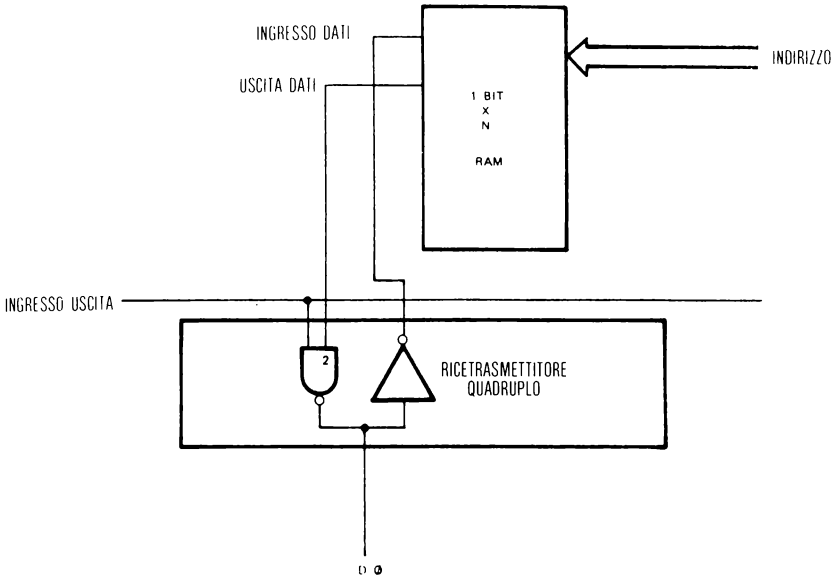


Fig. 2-16: RAM di memorizzazione temporanea delle vie dei dati e uso di un ricetrasmettitore di bus

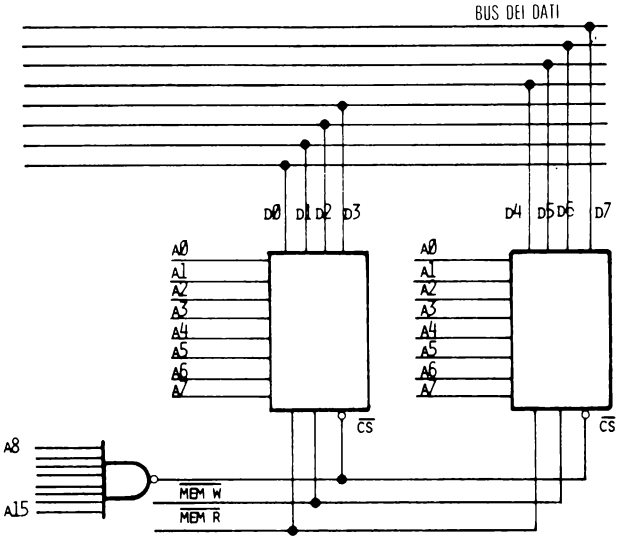


Fig. 2-17: Connessione della RAM 2111

Le vie del bus di indirizzo necessarie per specificare gli indirizzi sono connesse a ciascun modulo RAM. Le otto vie di indirizzo selezioneranno uno dei 256 byte in ciascun circuito integrato RAM. Le otto vie di indirizzo non utilizzate sono individuate da una porta NAND ad otto ingressi. Sulla base delle precedenti indicazioni, la RAM sarà localizzata nel campo di indirizzi compresi tra FF00 ed FFFF in esadecimale. Il bus dei dati è separato in due gruppi comprendenti 4 vie ciascuno. Ciascun gruppo di quattro bit trasferiti è caricato su una delle due RAM 256 per 4.

La figura 2-16 indica il modo nel quale è realmente realizzata la connessione.

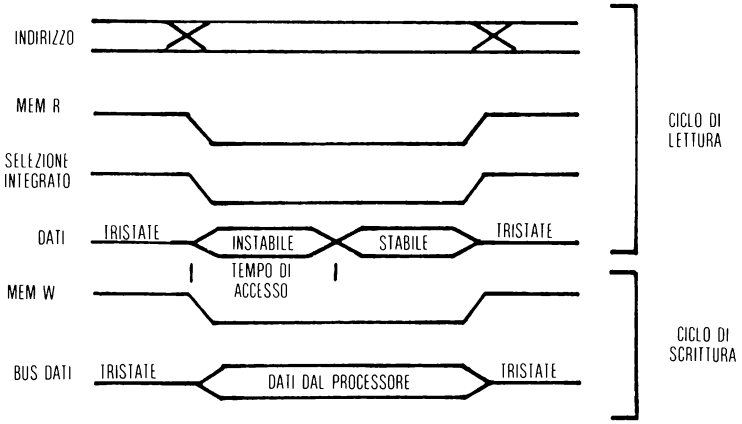


Fig. 2-18: Temporizzazione della RAM

Sono necessarie vie di controllo per abilitare le memorie alla lettura ed alla scrittura, oltre che per controllare la temporizzazione delle fasi di scrittura.

La RAM 2111 qui usata ha un certo numero di ingressi di abilitazione oltre a quelli di lettura-scrittura. I due segnali: «lettura memoria» e «scrittura memoria» sono usati per controllare le RAM «lettura memoria» abilita i circuiti di pilotaggio di uscita dal modulo per pilotare il bus dei dati. In tutti gli altri intervalli di tempo durante il quale tale pilotaggio non è attivato, il modulo è predisposto alla lettura, ma sul bus non sono presentate informazioni. «Scrittura memoria» abilita la RAM a realizzare un ciclo di scrittura e controlla l'apertura delle porte dei dati presenti nel bus dei dati verso la RAM. La figura 2-18 indica la temporizzazione di tali fasi.

Quando l'indirizzo è presente con i livelli di segnale stabilizzati e «lettura memoria» è tenuto a livello basso, il modulo è abilitato a pilotare il bus dei dati.

Dopo che il byte è stato individuato esso rimane sul bus finchè è prelevato dal processore e il livello di «lettura memoria» ritorna alto.

Il ciclo di scrittura è simile, eccetto che questa volta «scrittura memoria» è tenuto basso forzando la scrittura nella RAM del contenuto del bus dei dati.

L'integrazione del processore e della memoria entro un'area modulare del sistema richiede solo che essi siano indicati e interconnessi nello stesso schema.

Sistema 8080 completo

Per ottenere una maggiore espandibilità, il modulo di sistema contiene soltanto *decodifica parziale* per le PROM e *selezione lineare* per le RAM. Il modulo di memoria è indicato in figura 2-19. La PROM occupa le posizioni esadecimali tra 0000 ed 0FFF. La RAM sarà tra 2000 e 20FF esadecimale. Essa sarà inoltre indirizzata per tutti gli indirizzi della forma binaria 0XX1XXXXXXXXXXXXX e XX01111111111111 e in binario, dove X è un uno oppure uno zero (condizione di indifferenza). La PROM è indirizzata da tutti i valori delle forme binarie da XX00000000000000 a XX01111111111111.

Non possiamo aggiungere altra memoria a tale sistema senza ulteriore decodifica. Il processore centrale sarà lo stesso di quello indicato in figura 2-10. Come esercizio potrebbe essere esaminata la struttura del processore centrale del capitolo 8 e il lettore potrebbe verificare la propria corretta comprensione delle tecniche di decodifica di indirizzo e di memorizzazione temporanea.

Il sistema 6800

Impostato e progettato dalla Motorola il 6800 è un altro tipo di microprocessore «standard» e abbastanza diffuso. In confronto al componente della Intel, il 6800 presenta alcune differenze nella filosofia di progetto. La più macroscopica è l'assenza di multiplazione delle terminazioni e la presenza di una singola tensione di alimentazione. Altre differenze sono presenti nel set di istruzione, nell'architettura interna e nei segnali di controllo.

In generale, entrambi i componenti sono essenzialmente simili. La figura 2-20 indica una schematizzazione del sistema 6800.

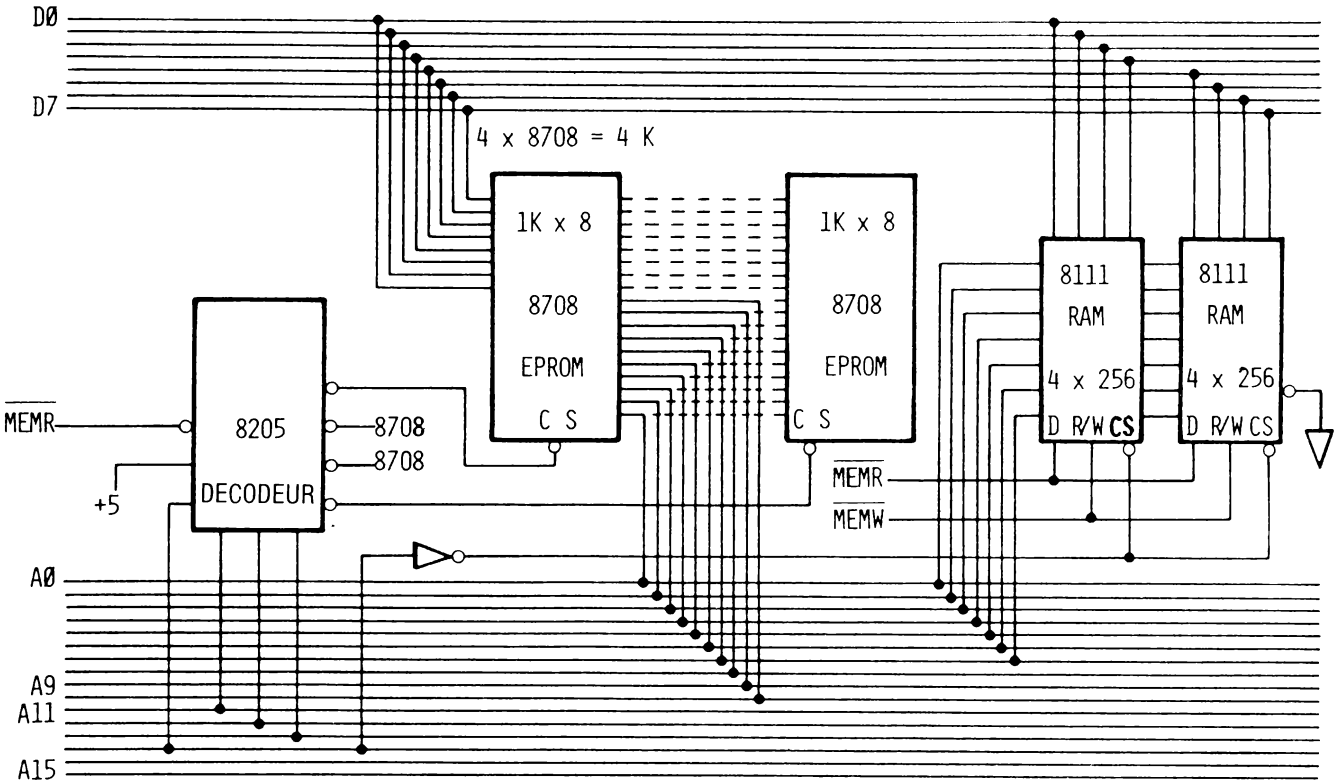


Fig. 2-19: Memoria completa del sistema 8080

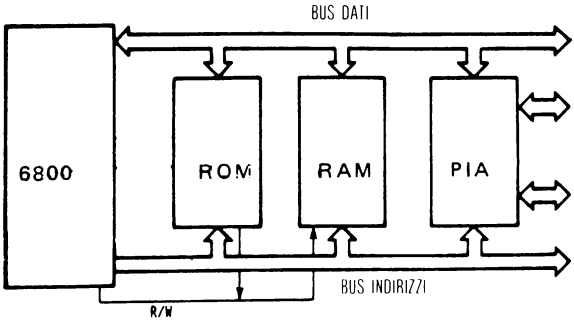


Fig. 2-20: Diagramma a blocchi del sistema 6800

Il segnale di temporizzazione

Il 6800 richiede un generatore di segnale di temporizzazione non TTL compatibile. Poichè il segnale a due fasi non svolge alcuna altra funzione, possono essere usati sia semplici circuiti descritti che generatori di segnale integrati. La Motorola produce un componente ibrido che contiene il cristallo e fornisce le due fasi del segnale necessario. In figura 2-21 sono indicate in dettaglio le caratteristiche del segnale.

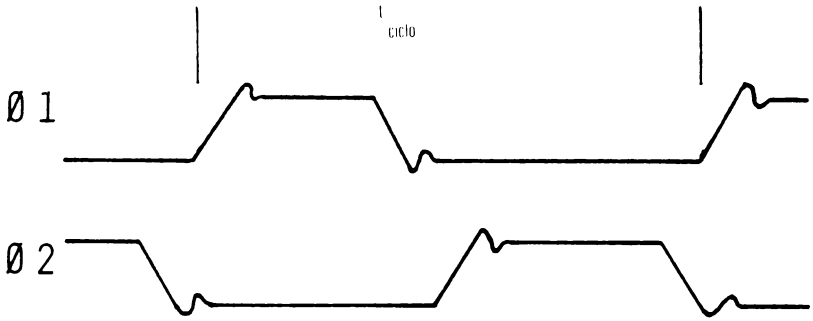


Fig. 2-21: Segnali di temporizzazione senza sovrapposizione del 6800

Bus del 6800

L'architettura del 6800 usa ingresso-uscita mappate in memoria (vedi capitolo 3) e richiede un unico livello di alimentazione, contro i tre livelli dell'8080. Questo comporta che non è necessario multiplare i segnali di controllo. Tuttavia i bus richiedono che i loro segnali subiscano una memorizzazione temporanea in sistemi di una certa dimensione, rendendo il numero totale dei pezzi essenzialmente eguale

sia nei sistemi con 8080 che in quelli con il 6800. (Il modulo 8228 controllore del sistema comprende infatti circuiti di pilotaggio dei bus dei dati).

Il bus dei dati è ad 8 bit bidirezionale. Esso richiede una memorizzazione temporanea per gran parte delle applicazioni. I componenti Motorola suggeriti per tale scopo sono quelli indicati in figura 2-22.

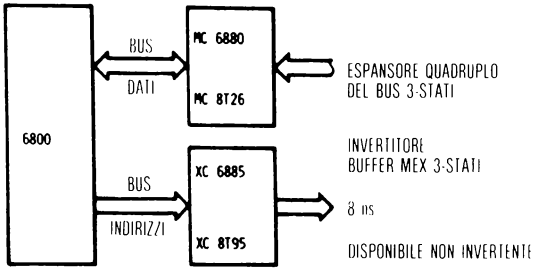


Fig. 2-22: Rigenerazione dei bus del 6800 - Componenti proposti

I bus di indirizzo e di controllo sono monodirezionali, con rispettivamente 16 vie di indirizzo e dieci vie di controllo. La figura 2-23 indica i segnali del bus del 6800. I segnali R/W, Φe VMA sono necessari per interfacciare la memoria. Essi servono rispettivamente per il controllo di lettura-scrittura, della fase due del segnale di temporizzazione e di validazione degli indirizzi di memoria.

- TSC — Il livello alto forza il bus di indirizzi e R/W nello stato ad alta impedenza
- DBE — Il livello basso forza il bus dati nello stato ad alta impedenza
- R/W — La MPU è nel modo/lettura quando il segnale è basso
- VMA — Indica indirizzo di memoria valido. Il livello alto abilita la RAM; il PIA e l'ACIA
- IRQ — È la via di richiesta di interruzione. Il PC è caricato da FFF8, FFF9
- RESET — Avvia il 6800 dopo uno stato di assenza di potenza. Il PC è caricato da FFFE, FFFF. Prima del reset è necessario siano eseguiti 8 cicli
- NMI — È una interruzione non mascherabile. Il PC è caricato da FFFC, FFFD
- HALT — Permette l'esecuzione del programma mediante una sorgente e un controllo dell'avanzamento esterni
- BA — (HALT o WAIT) indica che il bus degli indirizzi è disponibile.

Fig. 2-23: Segnale di controllo del 6800

La ROM

La Motorola produce una linea di prodotti compatibili con il 6800 che facilita le esigenze di interfaccia nei sistemi di piccole e medie dimensioni. La ROM di 1 K per 8 bit in tecnica di mascheratura comprende *quattro terminazioni di selezione del modulo* per selezionare la ROM, come indicato in figura 2-24.

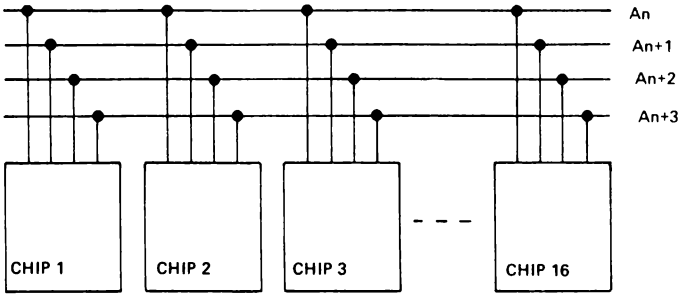


Fig. 2-24: La selezione di 4 moduli permette la connessione di fino a 16 moduli

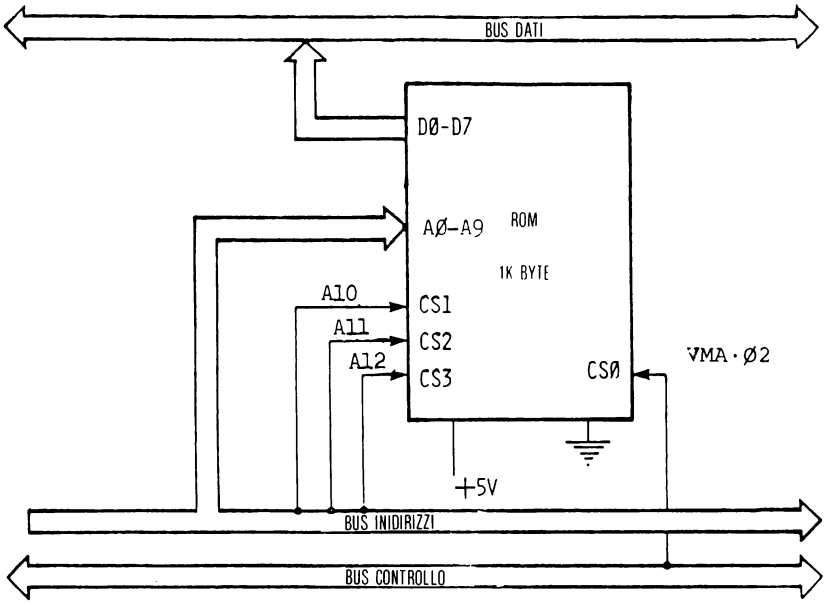


Fig. 2-25: Connessione della ROM del 6800

Nell'esempio di figura 2-25 i selettori di modulo sono connessi ai tre bit di ordine superiore del bus di indirizzo, ed al segnale VMA posto in AND con il segnale $\Phi 2$.

In tal modo, la ROM è selezionata per ogni ciclo di indirizzo di memoria valido da 1C00 a FFFF in esadecimale.

Naturalmente la ROM è solo di 1024 bytes, e l'ampio campo di indirizzi che essa impegna sono dovuti ai bit di indirizzo A15, A14, A13 non codificati ed a quelli non significativi.

La RAM

La Motorola è uno dei pochi costruttori che produce una RAM di 128 per 8 bit. Questa è una dimensione adatta per piccoli sistemi. Interfacciare verso la RAM 6810 è facilitato dal grande numero di selettori di modulo che sono presenti nel circuito.

L'interfaccia della RAM è indicata in figura 2-26. È da notare che sono necessarie soltanto sette vie di indirizzo per selezionare uno dei 128 bytes della RAM. Le altre nove vie di indirizzo possono essere usate in diverse combinazioni per selezionare il modulo. Nell'esempio di figura la RAM è selezionata quando le vie da A11 ad A7 sono tutte al livello basso. Ciò significa indirizzi esadecimali da 0000 a 00FF. Poichè i quattro bit di indirizzo più alti non sono interamente decodificati, la memoria è abilitata anche nei campi 2000 ÷ 20FF e così via, fino al campo F000 ÷ F0FF.

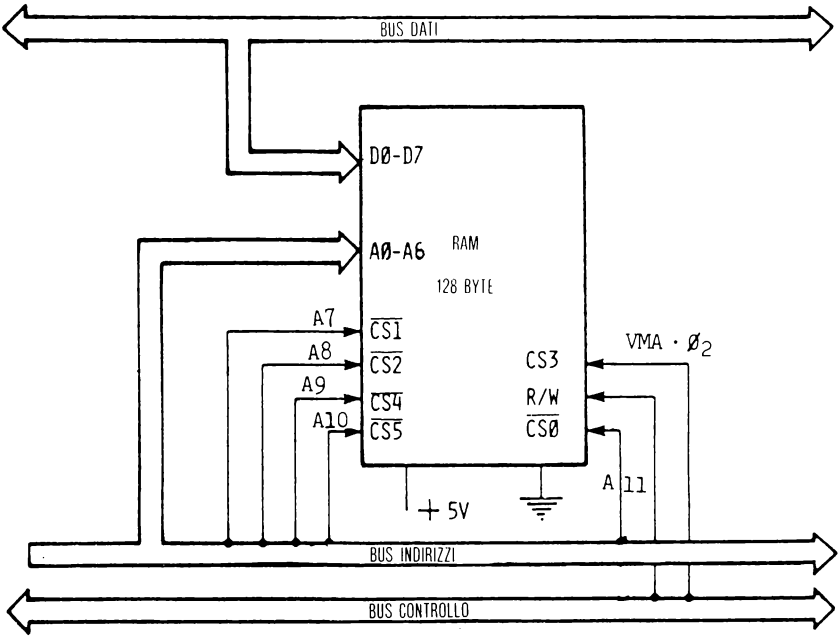


Fig. 2-26: Connessione della RAM del 6800 - Il 6810

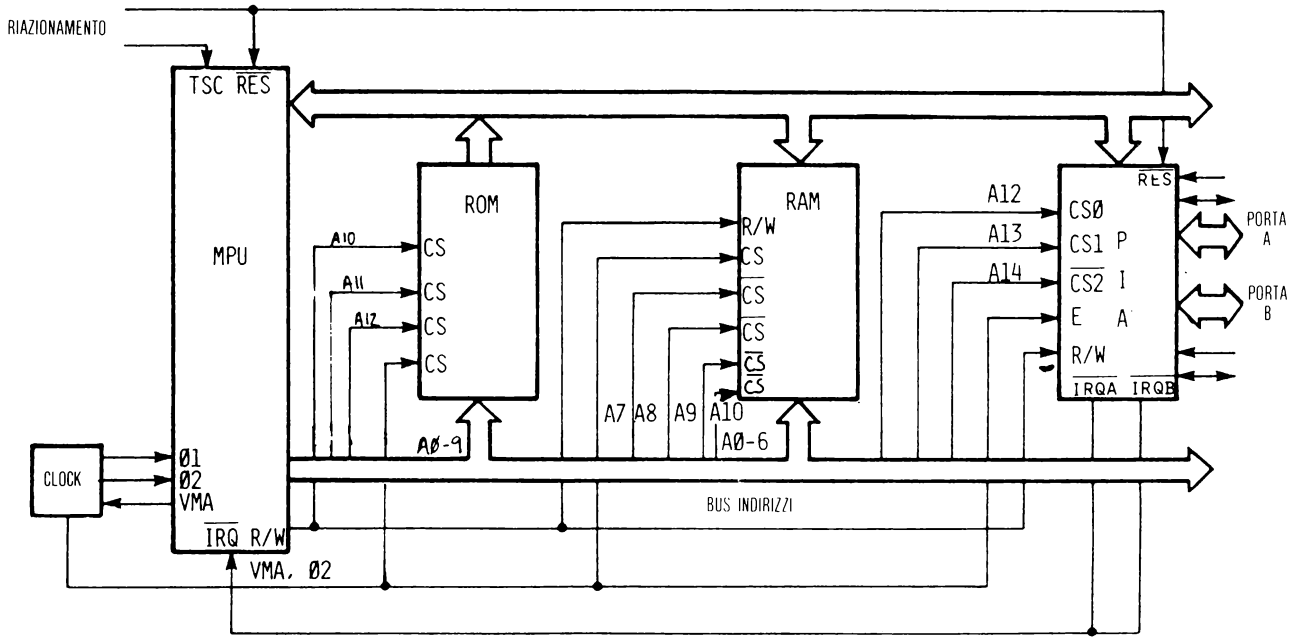


Fig. 2-27: Sistema 6800 completo

Per usare sia la RAM che la ROM correttamente è necessario indirizzare i due moduli in campi non sovrapposti. Un esempio è indirizzare la ROM da FC00 ad FFFF e la RAM da 0000 a 00FF.

I segnali VMA e Φ2 selezionano il componente durante il ciclo di memoria, mentre i controlli «lettura-scrittura» controllano le fasi di prelievo e memorizzazione.

Sistema 6800 completo

In figura 2-27 è indicato il sistema 6800 completo. In essa è rapppresentato anche un modulo di ingresso-uscita il cui uso sarà specificato nel terzo capitolo.

Lo Z-80

Quanto fino ad ora descritto faceva riferimento a processori sviluppati nello stesso periodo. La Zilog, fondata dagli stessi progettisti dell'8080, decise di incrementare l'efficienza del componente originale, e sviluppò un sistema compatibile nel software con l'8080 (nel nuovo modulo sono presenti alcune istruzioni addizionali e nuovi registri che incrementano la sua capacità elaborativa). In particolare lo Z-80 dispone di segnali necessari per interfacciare con componenti di *memoria dinamica,* caratterizzati da maggiore capacità.

Un piccolo sistema: realizzato con lo Z-80 è indicato in figura 2-28.

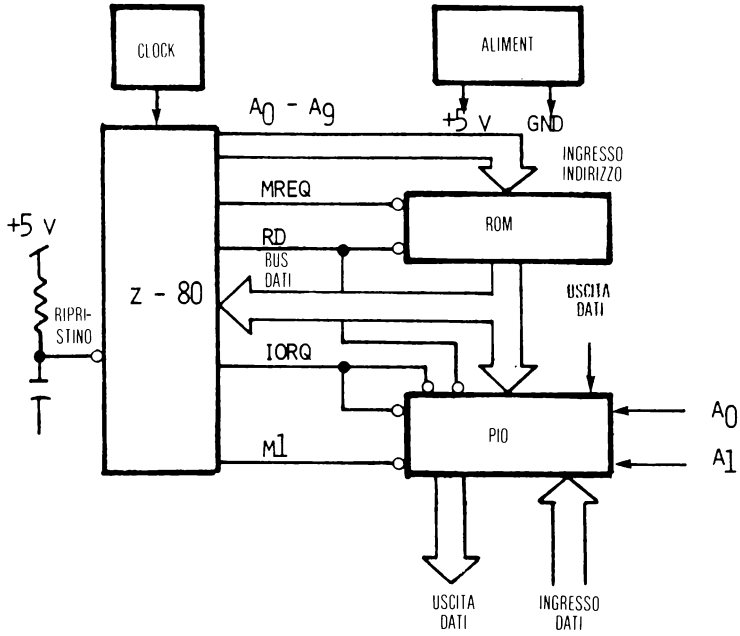


Fig. 2-28: Sistema Z-80

Interfaccia per RAM dinamica

Negli esempi precedenti la memoria usata era di tipo RAM statica. Con tali memorie l'informazione resta memorizzata finché è presente l'alimentazione; la RAM di *tipo dinamico* richiede invece un periodico *ripristino* dello stato delle celle poiché la memoria dinamica memorizza le informazioni in capacità di tipo FET. Tale componente è in grado di mantenere l'informazione solo per alcuni secondi. La cella deve essere indirizzata ad intervalli di alcuni millisecondi per ricaricare, o «rinfrescare», la cella. Lo Z-80 realizza questa funzione con un espediente di progetto.

Ogni qual volta è prelevata un'istruzione non è necessario che il bus di indirizzo mantenga una configurazione di indirizzo stabile. Invece di perdere tale tempo lo Z-80 presenta nei sette bit di indirizzo più bassi un *indirizzo di ripristino*. Questo indirizzo è incrementato ad ogni ciclo di istruzione. Con questo metodo di allocare un ciclo di ripristino ad ogni ciclo di istruzione e per la presenza di un addizionale registro di ripristino è stato possibile interfacciare facilmente le memorie dinamiche verso lo Z-80.

Altrimenti, il processore avrebbe dovuto attendere che un circuito dedicato, chiamato *controllore di ripristino,* facesse scorrere gli indirizzi lungo una riga di memoria dinamica per ripristinare le celle.

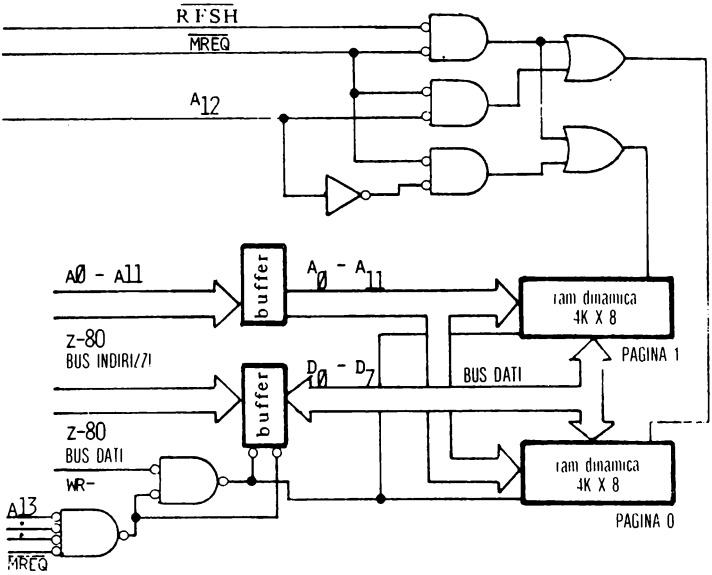


Fig. 2-29: Interfaccia della memoria dinamica verso lo Z-80

In figura 2-29, è indicata l'interfaccia della memoria dinamica.

In figura 2-30 è indicato il tipo di campionamento del ciclo di ripristino nello Z-80. In essa si può vedere come, ad ogni ciclo di istruzione, è intercalato un ciclo «libero» di ripristino.

Usando i segnali RFSH e MREQ è possibile realizzare un intervallo di ripristino preservando i dati.

In qualunque altro sistema sarebbe stato necessario prevedere un multiplatore di indirizzi, un contatore di fasi di ciclo e una logica di controllo del ripristino per ogni piastra di memoria. Un tale sistema è indicato in figura 2-31.

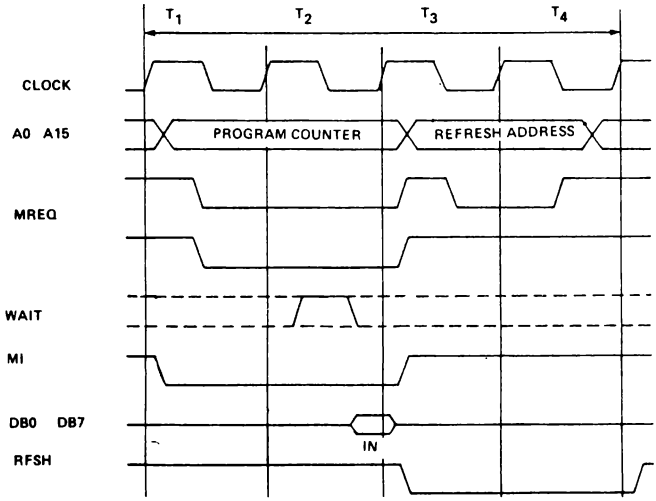


Fig. 2-30: Temporizzazione del ripristino nello Z-80

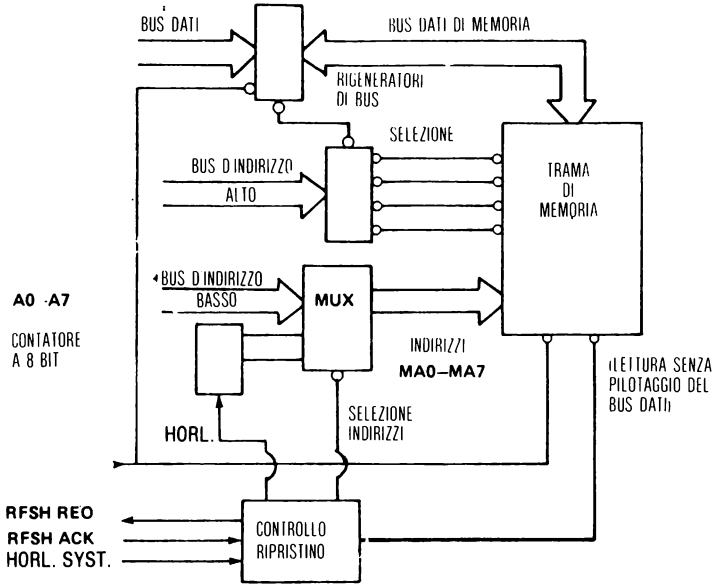


Fig. 2-31: Controllore di ripristino generale

La Sezione Controllo del Ripristino è differente per ciascun tipo di microprocessore in relazione alle specifiche di campionamento del bus. In accordo a tali specifiche deve essere scelto un opportuno metodo di ripristino.

La sezione di controllo può attendere due millisecondi e successivamente ripristinare tutte le colonne di celle o potrebbe, ad intervalli di un certo numero di istruzioni, ripristinare una colonna alla volta. L'ultimo metodo è preferibile perchè si adatta meglio alla temporizzazione dell'intero sistema. Per una più completa trattazione si può far riferimento a «Interfacciamento di Memorie RAM Dinamiche».

Uno schema efficiente è il *ripristino trasparente*. Questo è quanto lo Z-80 fa automaticamente. In generale, se è perfettamente conosciuta la complessa temporizzazione del bus, è possibile individuare un intervallo di tempo durante il quale la memoria non è usata. In tale intervallo un hardware opportuno può inserire un ciclo di ripristino.

La Mostek, che è il secondo fornitore dello Z-80, produce una CPU in una singola piastra, con 16 K bytes di RAM, 20 K bytes di ROM e un certo numero di porte di ingresso/uscita. La banca di RAM consiste di otto memorie dinamiche di 16 K ad un bit, mentre la banca di ROM consiste di cinque moduli a 4 K per un bit.

Questa singola piastra usa un limitato numero di circuiti integrati per implementare un efficiente processore. In confronto all'8080 la riduzione del numero dei moduli è dovuta all'eliminazione del modulo 8224 di temporizzazione, dell'8228 controllore di sistema e della logica di ripristino.

L'8085

Naturalmente anche la Intel ha migliorato il progetto dell'8080. L'8085, rispetto al suo antecedente, ha ridotto il numero di moduli e ha accresciuta la velocità. Essenzialmente esso comprende in un unico circuito integrato l'8080, l'8224 e l'8228.

Inoltre si è deciso di multiplare gli otto bit di indirizzo più bassi per fornire un accresciuto numero di funzioni di controllo, insieme alle sedici vie di indirizzo e alle otto vie di dati. All'inizio di ogni ciclo di istruzione sono predisposte sul bus dei dati le otto vie di indirizzo di ordine più basso.

È necessario uno stadio di controllo di trasferimento perchè esse possano essere usate. Per questo la via di controllo/multiplazione ALE («address-latch enable») assolve il compito di controllare il flusso e di trattenere il valore dei bit di indirizzo più bassi.

La figura 2-32 indica il sistema 8085. Da uno sguardo alla figura sembrerebbe che *non sia presente uno stadio di controllo-trasferimento per i bit di indirizzo più basso*. In realtà la Intel ha creato una nuova linea speciale di RAM, ROM, PROM e circuiti di ingresso/uscita che *contengono lo stadio di controllo di trasferimento dei bit di indirizzo di più basso ordine*. Pertanto l'8085 ha otto vie di dati, otto di indirizzo e undici vie di controllo.

I circuiti integrati periferici speciali contengono combinazioni di RAM, PROM e circuiti di ingresso/uscita. In tal modo un sistema completo può essere realizzato

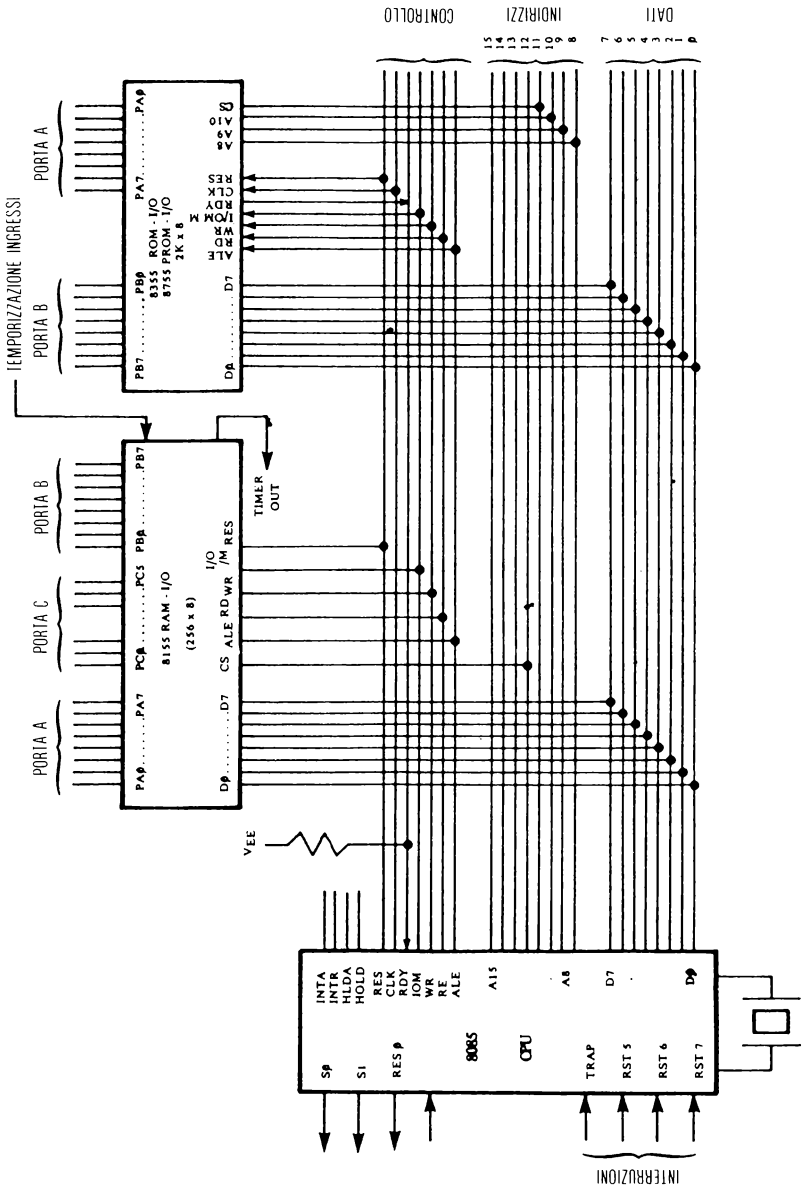


Fig. 2-32: Sistema 8085

con soltanto tre moduli LSI. In figura 2-33 è indicato il modulo 8277, PROM e I/O. La circuiteria di temporizzazione è stata inclusa nell'8085. Completa le interfacce verso la CPU la connessione di un cristallo a due specifiche terminazioni.

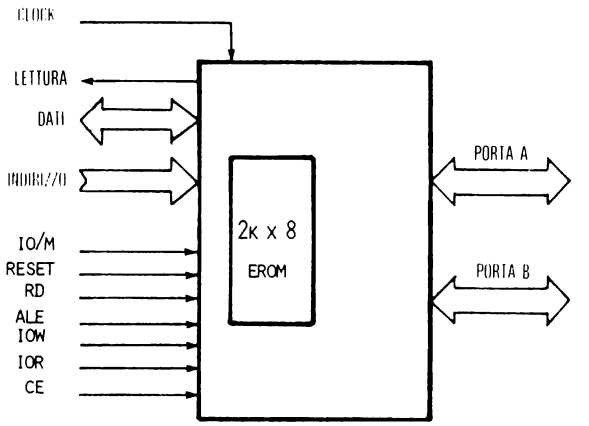


Fig. 2-33: PROM 8277 e I/O

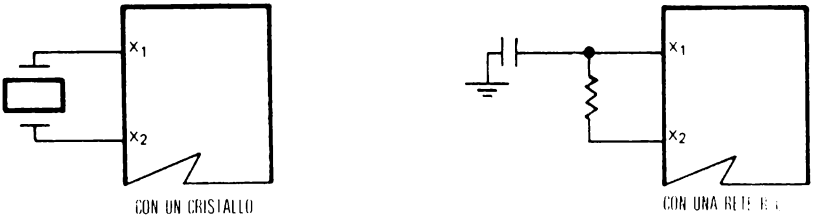


Fig. 2-34: Criteri di realizzazione del clock dell'8085

CONCLUSIONI

L'architettura standard del microprocessore, attraverso i suoi tre bus, controlla e caratterizza la struttura dell'intero sistema. I componenti di memoria, RAM e ROM, sono facilmente connessi ai bus standard del microprocessore. Piccoli sistemi usano la decodifica parziale o lineare per selezionare la memoria. Sistemi più grossi usano, invece, la decodifica completa degli indirizzi. Per illustrare la semplicità delle strutture di CPU sono stati considerati i sistemi 8080, 6800, Z-80 e 8085. I futuri processori conterranno ogni cosa indicata, eccetto il cristallo, rendendo le attuali strutture CPU superate.

Uniche funzioni che resteranno per completare un microcalcolatore saranno la memorizzazione temporanea dei segnali e l'interfacciamento per espletare le funzioni di I/O. Nel capitolo successivo saranno sviluppate le tecniche fondamentali di I/O, prima di indicare i criteri specifici delle periferiche attuali.

CAPITOLO 3

FONDAMENTI DI INGRESSO/USCITA

INTRODUZIONE

Adesso che la sezione di elaborazione del nostro microcalcolatore é completa, il passo successivo é comunicare con le periferiche. Le informazioni relative al mondo esterno devono essere selezionate e processate. Una volta processate, tali informazioni devono essere visualizzate e spedite per controllare i vari componenti. Questo capitolo sviluppa le tecniche di ingresso/uscita, e le illustra con esempi di progetto. Ciò sarà fatto in due fasi.

Saranno inizialmente descritte le tecniche fondamentali di ingresso/uscita: seriale o parallelo. Prima é introdotto il concetto; dopo é descritto il circuito integrato che implementa l'algoritmo.

Saranno successivamente descritte le tecniche di schedulazione necessarie per realizzare una corretta sequenza di ingresso/uscita: «polling», interruzioni e accessi diretti in memoria.

Sarà inizialmente chiarito un problema di terminologia. Elaboratori di maggior dimensione sono stati provvisti di istruzioni tipiche per colloquio con memorie o per I/O. *Questa distinzione non é più valida* per i microprocessori.

MEMORIE E MAPPE DI I/O NEI MODULI DI I/O

La implementazione tradizionale di elaboratori distingue istruzioni di I/O e di memoria.

I/O mappate come memorie

I/O mappate come memorie significa l'uso di istruzioni per interazione con la memoria per accedere a componenti di I/O. Tale tecnica permette al processore di usare le stesse istruzioni sia per trasferimenti da e verso la memoria che per operazioni di I/O. Una porta di I/O é trattata allo stesso modo di una posizione di memoria. Il vantaggio risultante consiste nella possibilità di usufruire della stessa efficienza delle istruzioni di scrittura e lettura in memoria per trasferire dati in ingresso e uscita. È però da osservare che in un elaboratore tradizionale sono eseguite normalmente più istruzioni di memoria che di I/O. Per esempio, con ingressi/uscite mappate come memoria, operazioni aritmetiche possono essere realizzate direttamente attraverso uno stadio di controllo di I/O, senza dover trasferire i contenuti in

registri temporanei.

Quali sono gli svantaggi? Anzitutto ciascuna porta di I/O usata in tal modo occupa una posizione disponibile per la memoria. Pertanto, se tutte le 65536 posizioni di memoria sono necessarie per essa, la tecnica di mappare come memoria gli I/O non può essere usata. Virtualmente, questo non dovrebbe capitare mai in un sistema a microprocessore. Inoltre, le istruzioni che operano sulla memoria richiedono normalmente tre bytes per indirizzare la posizione della porta (possono esistere 65536 posizioni, che richiedono 16 bit di indirizzo), mentre istruzioni speciali di I/O richiedono soltanto otto bit per specificare la porta. Infine le istruzioni di ingresso/uscita mappata in memoria possono richiedere maggior tempo di esecuzione di istruzioni speciali di I/O per la necessità di un numero maggiore di bytes di indirizzo. Questo problema é normalmente superato mediante l'indirizzamento «abbreviato», cioé mediante l'uso di istruzioni di memoria di due bytes.

Ingresso/uscita mappati come I/O

Nella tecnica di ingresso/uscita mappati come I/O il processore trasmette segnali di controllo indicanti che il ciclo in corso é solo per ingresso o uscita, non per memoria. Due vie speciali sono presenti per la scrittura e la lettura da I/O. Possono essere usate un numero minore di vie di indirizzo per selezionare la porta di I/O, poichè il sistema richiede un minor numero di porte di I/O che di posizioni di memoria.

Ci sono tre vantaggi ad usare l'ingresso/uscita mappati come I/O. Anzitutto, poiché sono usate istruzioni di I/O separate, esse possono essere distinte nella fase di programmazione da quelle che fanno riferimento alla memoria; questa é una comodità. Inoltre, per l'indirizzamento abbreviato, é necessario minor hardware per la decodifica. Infine, le istruzioni sono più corte. Gli svantaggi sono due: si perde in efficienza di elaborazione dell'ingresso/uscita mappati come memoria e, molto importante, si perdono due terminazioni di controllo per lettura e scrittura su I/O. Per

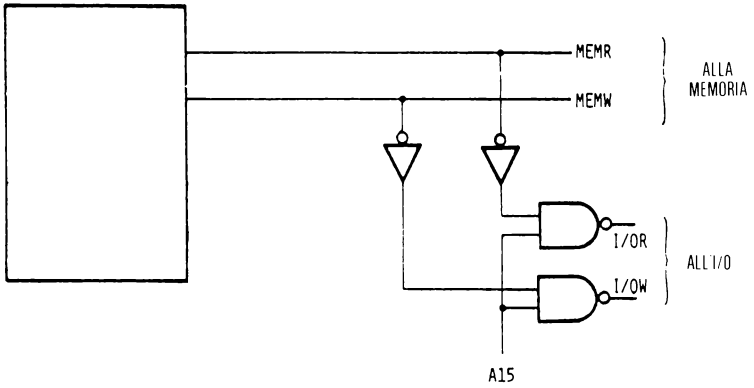


Fig. 3-1: Ingresso-uscita mappato in memoria

questa ragione questa tecnica non é quasi mai usata con i microprocessori (eccetto l'8080).

La figura 3-1 mostra un sistema di ingresso/uscita mappati come memoria. In esso il segnale di controllo che determina se l'indirizzo é per memoria o per I/O dipende dallo stato A15. Se esso é alto, tutti gli indirizzi definiti dal campo A14 - A0 specificano un componente di I/O. Se A15 é basso, lo stesso campo individua posizioni di memoria.

In figura 3-2 é indicato un sistema di ingresso/uscita mappati come I/O con vie di controllo separate per funzioni di controllo di memoria e di I/O. Il bus di indirizzo seleziona un componente e un registro o posizione entro il componente. Questo é illustrato in figura 3-4. Il bus di controllo specifica l'operazione che deve essere eseguita. Questo é il metodo di progetto standard in gran parte dei sistemi a microprocessore.

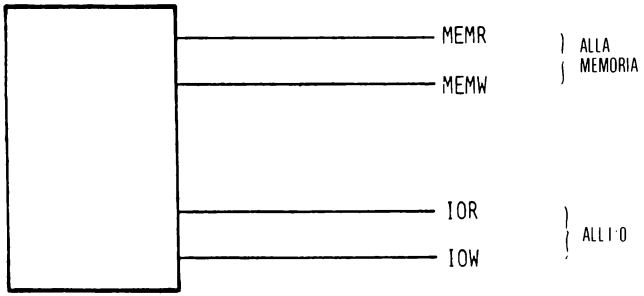


Fig. 3-2: Segnali di mappa di I/O

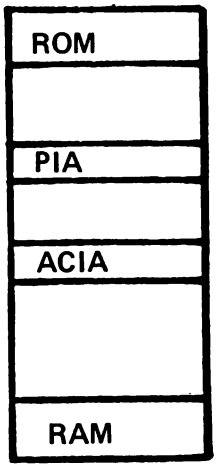


Fig. 3-3: Una mappa di memoria

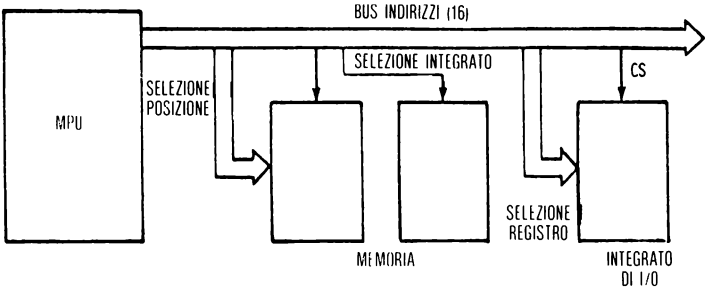


Fig. 3-4: Selezione di una porta di I/O

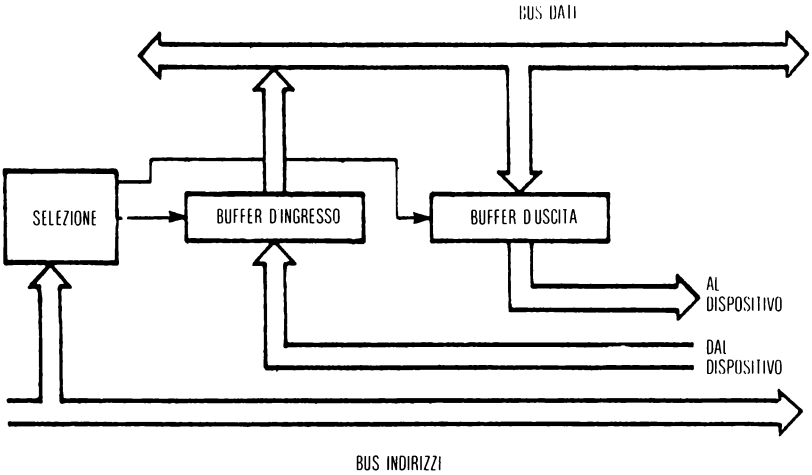


Fig. 3-5: Elementi basilari di una porta I/O

INGRESSO/USCITA PARALLELI

La più semplice interfaccia parallela richiede *controllori di trasferimento* e *circuiti pilota di bus*. Facendo riferimento ad una classica porta LSI di ingresso/uscita, in figura 3-5 una porta é provvista di una memoria ausiliaria di ingresso, che controlla il trasferimento del segnale di ingresso da un componente, e mantiene il suo stato stabile finché il microprocessore richiede l'informazione. Essa é anche provvista di una memoria ausiliaria di uscita per controllare il trasferimento dei dati del microprocessore, per mantere il loro stato finché il componente esterno li richiede. È necessario, inoltre, un meccanismo di *selezione* e un controllo lettura/scrittura per i registri o le porte. Le figure 3-6 e 3-7 illustrano quanto é richiesto da una semplice porta di I/O.

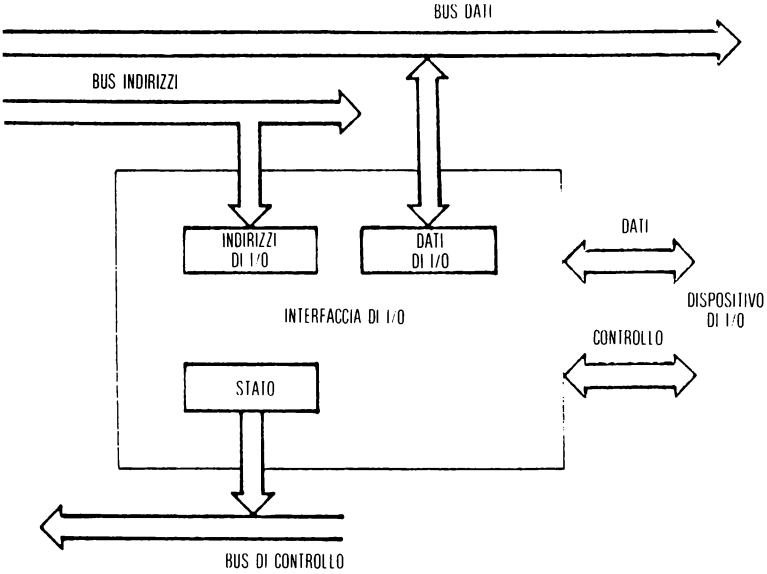


Fig. 3-6: Semplice porta di I/O

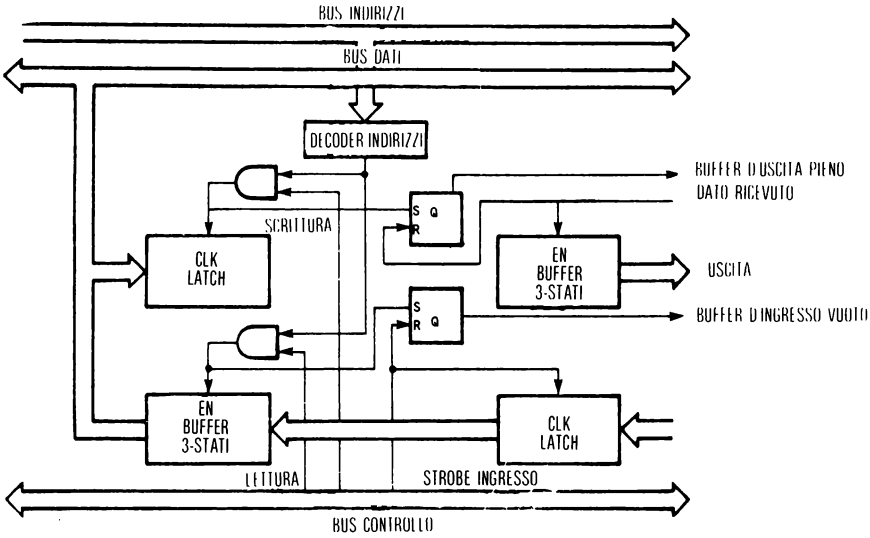


Fig. 3-7: Elementi di porta di I/O

Questo componente ha: un controllo di trasferimento per mantenere stabili i dati prodotti dal sistema fino alla loro uscita e memorie ausiliarie del bus per ricevere e pilotare il bus dei dati. Può anche essere presente un registro interno di stato indicante se ci sono dati da leggere o se i dati sono in uscita. Anche se tali porte possono essere realizzate con componenti discreti, un nuovo circuito, il PIO, ha reso tale metodo superato.

Componente integrato programmabile per ingresso/uscita parallelo

Il componente integrato programmabile per ingresso/uscita parallelo (PIO) svolge le seguenti funzioni: decodifica di indirizzo, memorizzazione temporanea e multiplazione dei dati in ingresso/uscita, indicazione e decodifica di stato per una corretta sequenza di scambio («handshaking»), ed altre funzioni di controllo che saranno indicate.

Il decodificatore di indirizzo seleziona uno dei registri interni sul quale operare la lettura o la scrittura. Questi registri possono essere di tipo per controllo ingresso e uscita, registri di direzione o registri di stato. Normalmente, per un numero di registri interni da sei ad otto sono necessari tre bit di indirizzo, oltre al selettore di circuito. Infine, *il PIO é «programmabile».*

Ciò é possibile per l'uso di un «registro di direzione dati»: la porta é definita, a livello di bit, come una entità che abbia i primi tre bit configurati come ingresso e i restanti cinque come uscita, o una qualsiasi altra combinazione. La direzione di ciascuna via del del PIO é cioé programmabile. Ciascun bit del «registro di direzione dei dati» specifica se il corrispondente bit della porta del PIO é un ingresso o una uscita. Tipicamente, uno «0» di un bit del registro di direzione dei dati specifica un ingresso, mentre un «1» specifica un'uscita. Un PIO é programmabile in altri modi. Ciascun PIO ha uno o più registri di comando che specificano le opzioni, come la configurazione delle porte e il modo di operare della logica di controllo.

Infine ciascun PIO multipla le sue connessioni al bus dati del microprocessore su due o più porte di otto bit. Il massimo é tre, includendo le vie di controllo per il componente di I/O, per la limitazione di 40 terminazioni del contenitore. Un PIO tipico é indicato in figura 3-8. In tal caso il componente ha due porte, ciascuna provvista del proprio registro di direzione. Inoltre, é usato un registro di stato per indicare lo stato di ciascuna porta.

Esempio 1: il PIA 6820 della Motorola

Il diagramma interno del 6820 é indicato in figura 3-9. Esso ha sei registri, un gruppo di tre registri per porta. Un gruppo é per la porta A e l'altro per la porta B.

Esaminiamo il registro di controllo. Il suo formato é indicato in figura 3-10. Il bit 7 indica una transizione dell'ingresso CA1. Esso é usato come un semaforo di interruzione. La stessa cosa vale per il bit sei, eccetto che esso é associato alla terminazione CA2 usata come ingresso. I bit 5,4 e 3 fissano gli otto differenti modi di funzionamento del componente e la funzione della terminazione CA2.

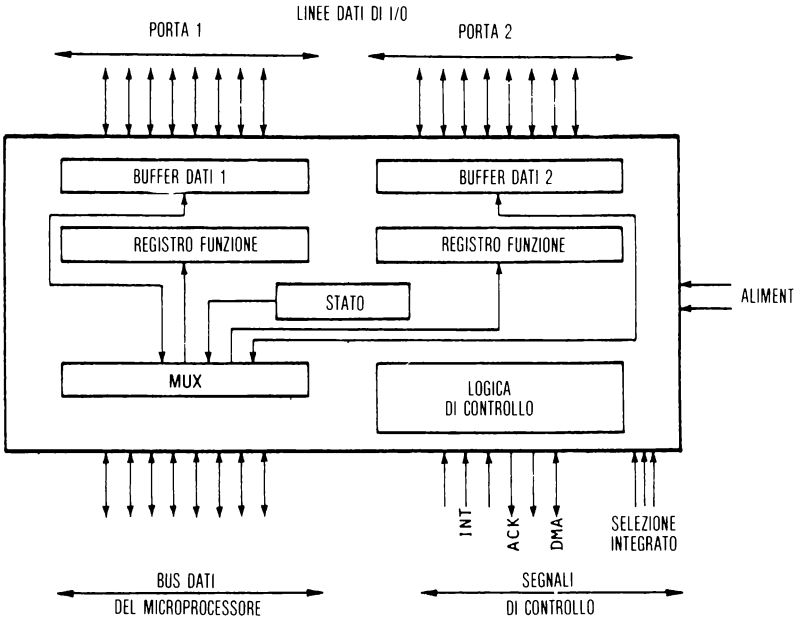


Fig. 3-8: PIO tipico

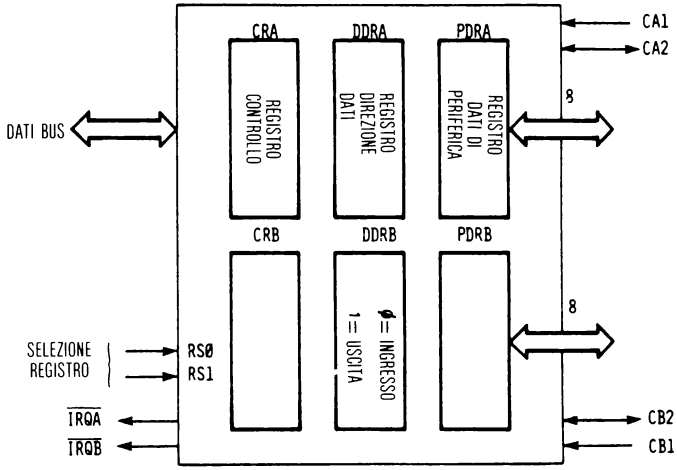


Fig. 3-9: PIA 6820

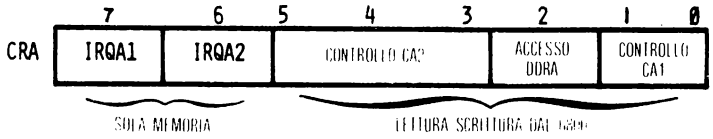


Fig. 3-10: Formato del registro di controllo nel 6820

Il bit 2 indica se deve essere selezionato il registro di direzione o il registro dati, poiché essi hanno lo stesso indirizzo. I bit 1 e 0 sono di controllo per abilitazione e disabilitazione dell'interruzione.

A questo punto é necessaria una chiarificazione: il PIA della Motorola ha sei registri e soltanto due terminazioni per selezionare registri (RS) a causa della limitazione di 40 punti di connessione del contenitore. Il DR e il DDR in ciascuna porta condividono lo stesso indirizzo. Essi sono differenziati dal bit 2 del registro di controllo, una cattiva soluzione di programmazione.

La figura 3-11 indica come i registri sono selezionati mediante l'uso delle terminazioni RS1 e RS0, e lo stato del bit interno 2 del registro di controllo.

La selezione dei registri della PIA usa due vie (RS0 e RS1) oltre al bit 2 di CR:

RS1	0	Seleziona il registro della porta A
RS1	1	Seleziona il registro della porta B
RS0	1	Seleziona il registro di controllo (A o B)
RS0	0	Seleziona il registro di direzione o la memoria tampone

RS1	RS0	CRA (2)	CRB (2)	REGISTRO	
0	0	0	–	Registro direzione dati	A
0	0	1	–	Registro della memoria tampone	A
0	1	–	–	Registro di controllo	A
1	0	–	0	Registro direzione dati	B
1	0	–	1	Registro della memoria tampone	B
1	1		–	Registro di controllo	B

Fig. 3-11: Selezione dei registri nel 6820

La figura 3-12 indica la connessione del PIA ai bus del 6800 e la figura 3-13 illustra una tipica applicazione evidenziando la codifica dei bit dei registri di controllo e di direzione.

Ultima osservazione sul 6820 e che é una buona idea usare una memoria temporanea per tamponare il bus dei dati verso il modulo, visto che esso non riesce a pilo-

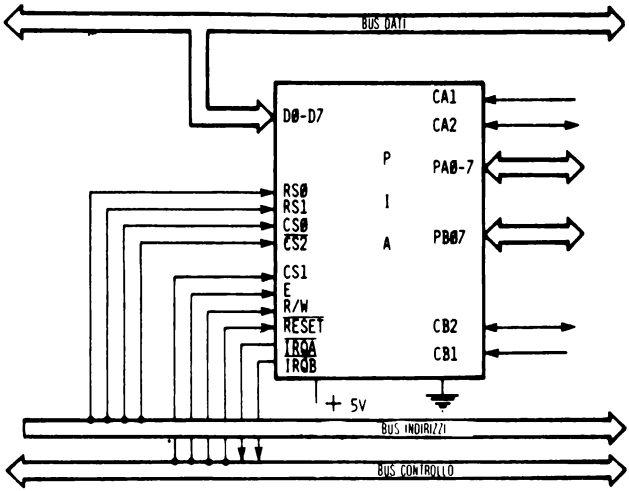


Fig. 3-12: Interfaccia tra il 6820 e il 6800

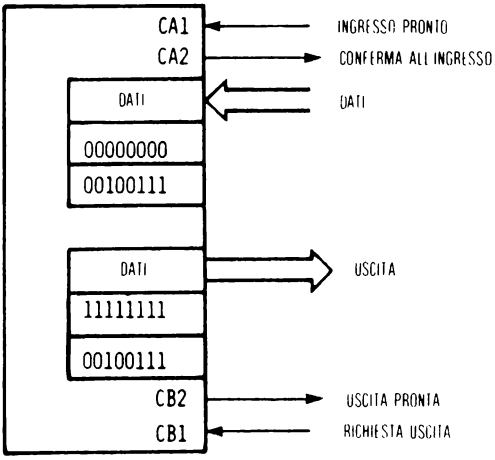


Fig. 3-13: Applicazione del 6820

tare a sufficienza se é abbastanza caricato. La figura 3-14 propone una soluzione circuitale adatta allo scopo.

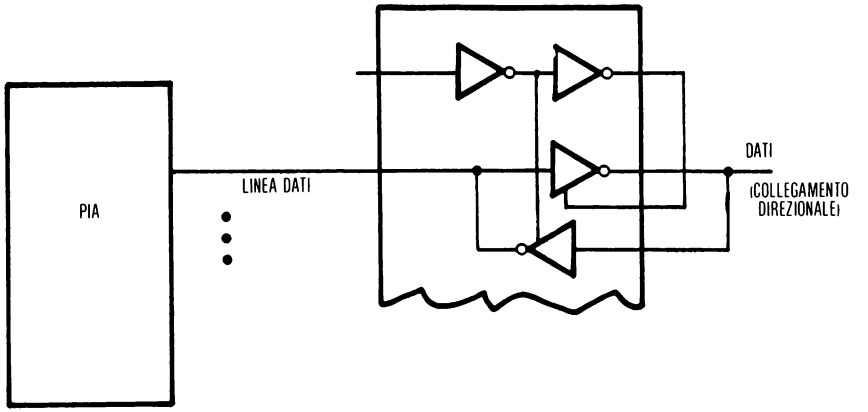


Fig. 3-14: Rigenerazione del bus dati

Esempio 2: PPI 8255 Intel

Lo 8255 contiene quattro porte, due con 8 bit ciascuno e due con quattro. Ciascuna porta può essere programmata mediante il registro di controllo di modo, per funzionare insieme a tutte le altre come ingresso o come uscita o per realizzare una funzione speciale. La figura 3-15 mostra la struttura dell'8255. La figura 3-16 mo-

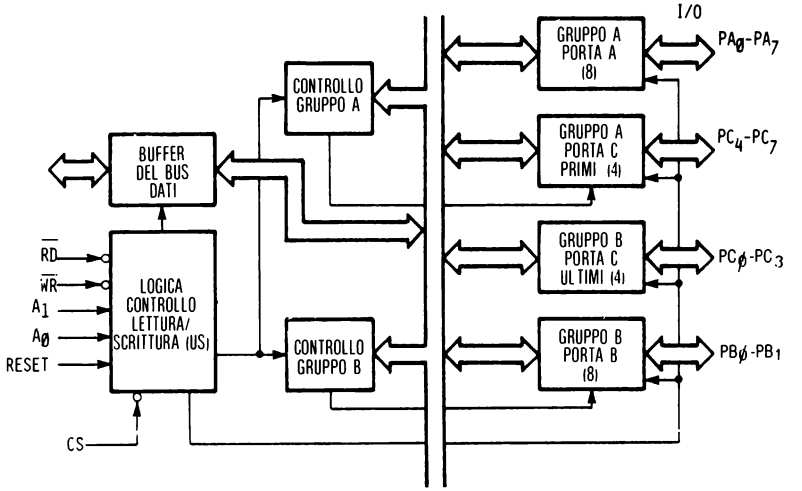


Fig. 3-15: Indirizzamento dell'8255

stra, invece, come sono indirizzate le porte. Ci sono diversi modi di funzionamento, nei quali ciascuna metà della porta C é usata come ingresso di semaforo di interruzione o segnale di controllo sequenza («handshaking»). Il componente Intel non é programmabile mediante bit, ma offre 4 vie addizionali per controllo. In generale, le funzioni realizzate sono essenzialmente simili. Infatti un PIA può essere usato con un sistema 8080 e viceversa. Ciascun principale costruttore di microprocessori ha la sua versione della interfaccia programmabile parallela. La funzione é essenzialmente la stessa.

$\overline{CS}$	A1	A0	$\overline{RD}$	$\overline{WR}$	OPERAZIONE	
0	0	0	0	1	Porta A al bus dati	lettura di MPU (A,B,C)
0	0	1	0	1	Porta B al bus dati	
0	1	0	0	1	Porta C al bus dati	
0	0	0	1	0	Bus dati alla porta A	scrittura di MPU
0	0	1	1	0	Bus dati alla porta B	
0	1	0	1	0	Bus dati alla porta C	
0	1	1	1	0	Bus dati a controllo	
0	1	1	0	1	Illegale	
1	—	—	—	—	Bus dati a controllo tri-state (disabilita)	

Fig. 3-16: Indirizzamento dell'8255

INGRESSO/USCITA SERIALE

Svariati componenti richiedono una comunicazione seriale: teletype (TTY), nastro magnetico e disco.

Invece di controllare il trasferimento di otto bit di dati in parallelo é possibile trasferire otto bit entro un byte, uno alla volta, attraverso una singola via. Questa tecnica, conosciuta come interfaccia seriale di bit, dispone di procedure standard di trasmissione. Tali standard sono discussi nel capitolo 6. Il formato di ingresso uscita seriale verso una teletype é indicato in figura 3-17.

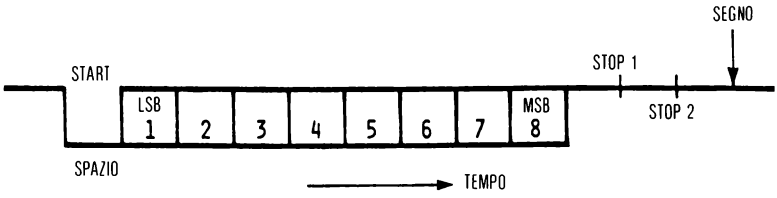


Fig. 3-17: Formato del carattere seriale

Poiché i microcalcolatori sono sistemi paralleli, é necessario fare una conversione parallelo/serie per disporre dei dati in uscita, e viceversa per l'ingresso. Esistono due metodi per realizzare questa conversione: mediante software o mediante un UART (Universal - asynchronous receiver - transmitter).

I/O seriale mediante software

Un programma può facilmente realizzare in software la serializzazione e viceversa. In ingresso il programma attenderà, finché esso non rivelerà un bit di start, per campionare in tempi opportuni per leggere i bit dati. In uscita, il programma trasmette serie di uni e zeri verso una singola linea, con un ritardo programmato tra un bit e l'altro.

Un esempio di programma di uscita su teletype é indicato nel diagramma di flusso della figura 3-18, mentre la lista di programma dell'8080 é indicata nella figura 3-19.

Essa sarà descritta nel capitolo 4. Il principio di una routine di serializzazione é di assemblare 8 bit (o più), costituenti la parola dell'accumulatore e di far slittare tali bit, uno alla volta, alla frequenza opportuna. Il metodo più semplice é di presentare il contenuto dell'accumulatore su una porta di uscita che é connessa solo alla via zero. L'accumulatore é successivamente fatto slittare a destra di un bit, é programmato un ritardo, e il bit successivo é presentato in uscita. Dopo 8 (o più) uscite, i dati inizialmente in parallelo sono stati serializzati.

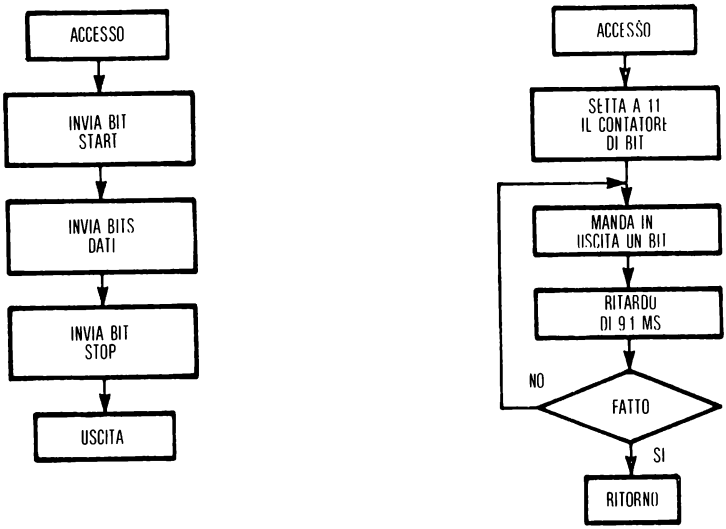


Fig. 3-18: Diagramma di flusso per la conversione seriale

```
;
;  THIS SUBROUTINE ENTERED WITH CHARACTER TO BE OUTPUT IN THE C REGISTER
;
TYOUT:   MVI    B,11     ;  SET COUNTER FOR 11 BITS
         MOV    A,C      ;  CHARACTER TO ACCUMULATOR
         ORA    A        ;  CLEAR CARRY-FOR START BIT
         RAL             ;  MOVE CARRY TO A(0)
MORE:    OUT    2        ;  SEND TO TTY
         CALL   DELAY    ;  KILL TIME
         RAR             ;  POSITION NEXT BIT
         STC             ;  SET CARRY-FOR STOP BITS
         DCR    B        ;  DECREMENT BIT COUNTER
         JNZ    MORE     ;  DONE?
         RET             ;  YES
;
;  9 MSEC DELAY (ASSUME NO WAIT STATES)
;
DELAY:   MVI    D,6
DL0:     MVI    E,2000
DL1:     DCR    E        ;  1.5 MSEC
         JNZ    DL1      ;  INNER LOOP
         DCR    D
         JNZ    DL0
```

Fig. 3-19: Programma di conversione seriale per l'8080

Inversamente, raccogliere bit in serie per una conversione in parallelo via software é abbastanza semplice. Il bit 0 é letto nell'accumulatore. L'accumulatore é fatto scorrere a destra. Dopo un ritardo prefissato é letto nuovamente il bit 0. Dopo 8 scorrimenti il byte risulta assemblato.

Il vantaggio di un'implementazione programmata é la semplicità e l'eliminazione di hardware esterno. Tuttavia essa é lenta e riduce l'efficienza del microprocessore. Inoltre non é possibile realizzare un ritardo preciso in un sistema che usa interruzioni. È richiesta una realizzazione hardware.

UART e USART

Uno dei primi componenti LSI standard é stato lo UART. Lo UART é un convertitore serie/parallelo e parallelo/serie. Esso ha due funzioni: prelevare dati in parallelo e convertirli in una sequenza seriale con caratteri di start, parità e stop; ricevere una sequenza seriale di bit e convertirla in dati in parallelo.

In figura 3-20 é indicato il diagramma funzionale a blocchi dell'UART. Ciascun UART ha tre sezioni: un trasmettitore, un ricevitore e una sezione di controllo. Quasi tutti i costruttori hanno una versione compatibile a livello di terminazioni o una versione «più efficiente» dell'UART standard.

Lo UART richiede sia una porta di ingresso che una di uscita per interfacciare un sistema a microprocessore, e pertanto gli UART sono stati progettati con bus direttamente compatibili con quelli del microprocessore. Due esempi sono: il Motorola MC6850 ACIA (asynchronous communication interface adaptor), e l'Intel USART (universal synchronous and asynchronous receiver - transmitter).

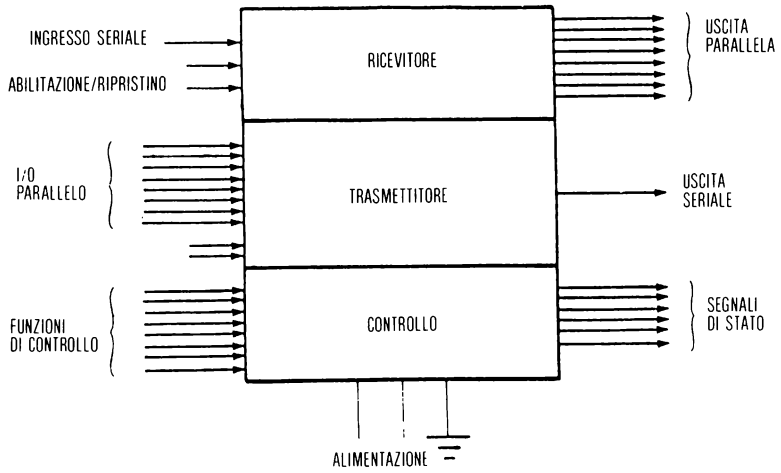


Fig. 3-20: Diagramma a blocchi dell'UART

Esempio 1: il Motorola 6850 ACIA

In figura 3-21 é indicato il diagramma a blocchi dell'ACIA. Oltre ai registri di ingresso e uscita serie/parallelo, la circuteria di controllo realizza le funzioni di controllo dello standard EIA RS232C (vedere il capitolo 6 per i dettagli della interfaccia RS232C).

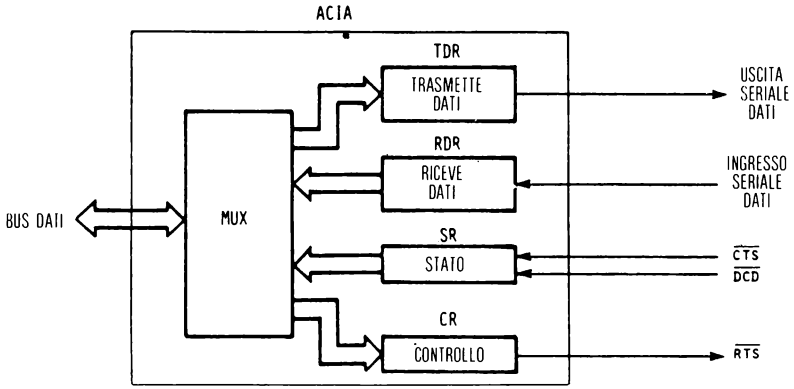


Fig. 3-21: ACIA 6850

La figura 3-22 divide gli ingressi e le uscite nelle loro funzioni: i dati seriali, il controllo del modem, i segnali di temporizzazione (i clocks) ed i bus. I dati seriali in ingresso e uscita sono TTL-compatibili e devono essere caricati su registri ausiliari per fornire i livelli necessari per pilotare i circuiti (vedere il capitolo 4 per una dettagliata indicazione su come connettere una teletype a un ACIA). I controlli del modem controllano la interfaccia richiesta in una connessione modem RS232C. Il segnale di temporizzazione controlla la velocità di linea dei dati seriali e può essere differente per la sezione di ricezione e di trasmissione. I segnali del bus sono i segnali usati nel sistema 6800. In figura 3-23 è riportata la «tabella della verità» dell'indirizzamento dei registri interni.

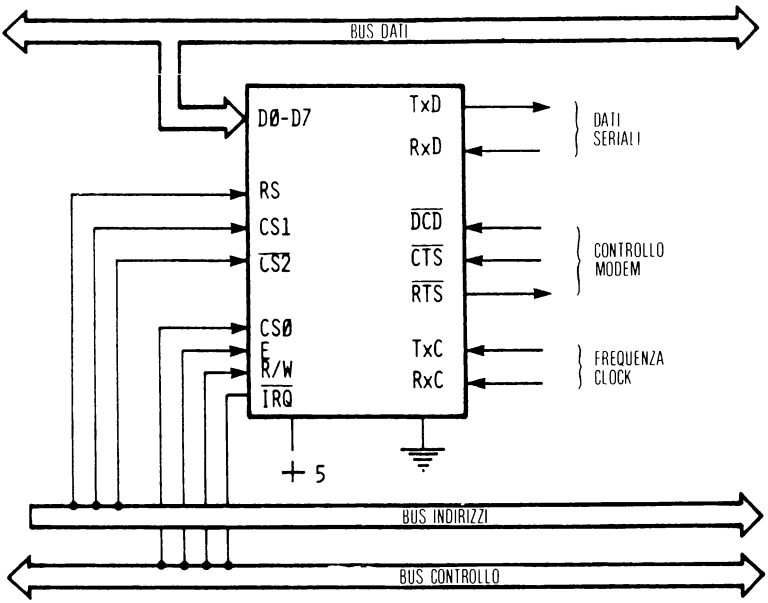


Fig. 3-22: Funzioni dell'ACIA 6850

RS	R/W	REGISTRO
0	0	Controllo
0	1	Stato
1	1	Ricevi dati
1	1	Trasmetti dati

Fig. 3-23: Indirizzamento dei registri interni del 6850

Esempio 2: Intel 8251 USART

Il diagramma a blocchi e i segnali di controllo dell'8251 USART sono indicati in figura 3-24. Questo componente differisce dall'ACIA: esso fornisce anche trasmissione e ricezione *sincrona*, oltre alla trasmissione asincrona (la Motorola dispone di un diverso USART lo «SSDA» per la comunicazione sincrona). La interfaccia dell'8251 verso il sistema 8080 è indicata in figura 3-25. Parte della circuiteria interna dell'8251 è dinamica, e pertanto essa richiede il segnale di temporizzazione Φ2. Il resto dei segnali hanno un significato facilmente comprensibile. La USART ha cinque registri interni: ricezione dati, trasmissione dati, modo, stato e controllo.

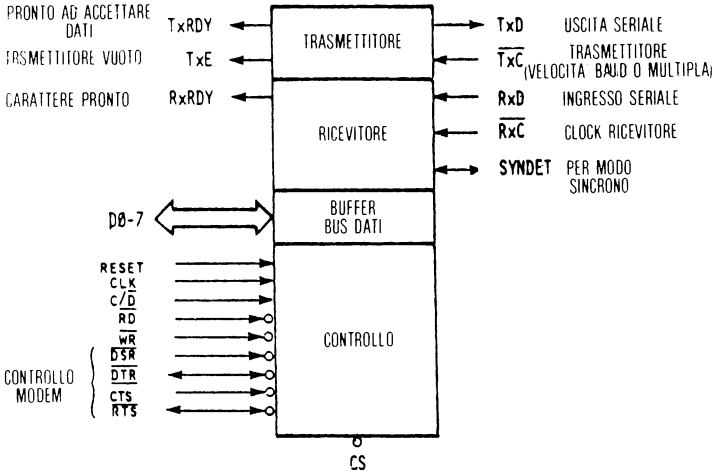


Fig. 3-24: USART 8251

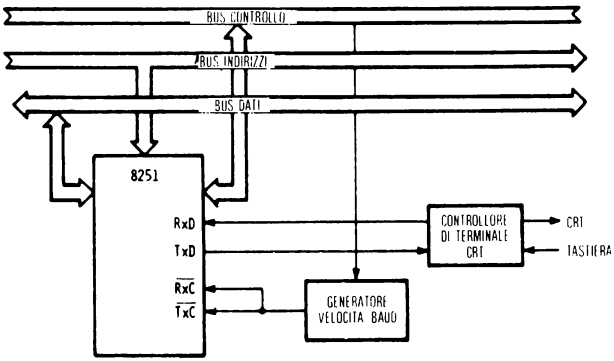


Fig. 3-25: Interfaccia tra l'8251 e l'8080

Dopo il «reset», il primo byte trasmesso all'8251 come controllo determina il *modo*. Il byte successivo trasmesso come controllo sarà considerato un *controllo*. Il *modo* determina se l'8251 deve essere usato in modo sincrono o asincrono. Il *controllo* indica la lunghezza della parola e altri parametri. La figura 3-26 è la tabella della verità del bus di controllo dell'8251.

$\overline{C/D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	OPERAZIONE
0	0	1	0	8251 verso bus dati (lettura)
0	1	0	0	bus dati verso 8251 (scrittura)
1	0	1	0	stato verso bus dati
1	1	0	0	bus dati verso controlli
--	--	–	1	bus dati verso tri-state

Fig. 3-26: Tabella di verità dell'indirizzamento dell'8251

Conclusioni sulle interfacce seriali

I due metodi presentati, hardware e software, sono un esempio della tradizionale alternativa nella decisione da prendere, anche nel più semplice progetto di interfaccia. Gran parte dei piccoli sistemi usano una interfaccia seriale software mentre, sistemi più grossi tendono ad usare lo UART. Circuiti ancora più sofisticati stanno per esser introdotti per realizzare nuovi tipi di comunicazione sincrona e asincrona. Questi componenti LSI realizzano gli altri standard seriali descritti nel capitolo 6.

I TRE METODI DI CONTROLLO DELL'INGRESSO/USCITA

Sono state già introdotte le tecniche e i componenti necessari per le interfacce fondamentali di I/O: possono essere realizzate porte seriali e parallele. Il problema successivo è come gestire il trasferimento dei dati, cioè, come realizzare una *strategia di schedulazione*. Sono usate tre tecniche fondamentali che sono di seguito brevemente descritte. Circuiti integrati addizionali saranno descritti, necessari per facilitare ciascuna delle tre strategie. I tre metodi sono raffigurati in figura 3-27. Essi sono chiamati: «polling», controllo delle interruzioni, e accesso diretto in memoria (DMA) (Sono usate anche combinazioni di esse).

I/O programmato o Polling

Nell'ingresso/uscita programmato tutti i trasferimenti verso e dai componenti sono realizzati mediante programma. Il processore spedisce e richiede dati: tutte le operazioni di ingresso/uscita sono sotto il controllo del programma da far girare. Il trasferimento deve essere coordinato da un processo per scambio ordinato («Handshaking»). Il metodo fondamentale per determinare se è richiesta un'operazione di I/O consiste nell'uso di *semafori*. Un semaforo è un bit che, quando è uguale ad u-

no, indica che si è verificata una condizione che richiede attenzione. Per esempio, un semaforo indica «componente pronto» = memoria di accesso pronta per un componente di ingresso, o memoria di accesso vuota per un componente di uscita. Il semaforo è controllato ad intervalli di tempo: questo è il «polling». La caratteristica di questo metodo è l'uso di una quantità minima di hardware, a spese di un overhead del software. Un diagramma di flusso di un *anello di polling* è indicato in figura 3-28. Il programma scorre continuamente attraverso una serie di test per determinare se l'ingresso/uscita può/potrebbe essere realizzato. Quando è individuato un componente che richiede servizio, è attivata una opportuna routine di servizio e il polling riprende dopo il completamento. Due metodi fondamentali sono utilizzati per individuare i semafori dei componenti pronti: l'uso di una semplice porta di stato dell'ingresso, e l'uso di una porta di stato di ingresso codificatrice di priorità. La tecnica più semplice è quella di pilotare il bus dei dati con i semafori per componenti pronti relativi ad otto di essi quando è eseguita un'istruzione di lettura di stato delle porte di ingresso. La figura 3-29 indica un tale sistema. La porta di stato di ingresso può essere un decodificatore di indirizzo. Normalmente è usato il primo o l'ultimo indirizzo di porta per tale uso. Quando la porta è letta in stato on, il programma controlla il livello di ciascun bit, determina la priorità, e passa all'esecuzione della routine opportuna.

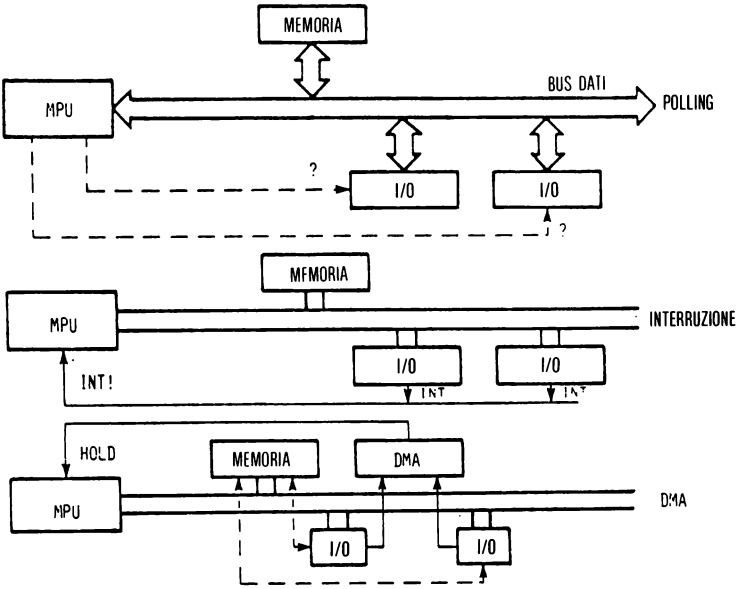


Fig. 3-27: Tre metodi di controllo di I/O

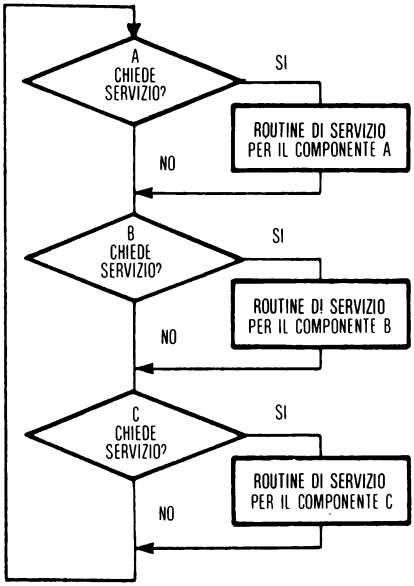


Fig. 3-28: Diagramma di flusso di polling circolare entro la lista

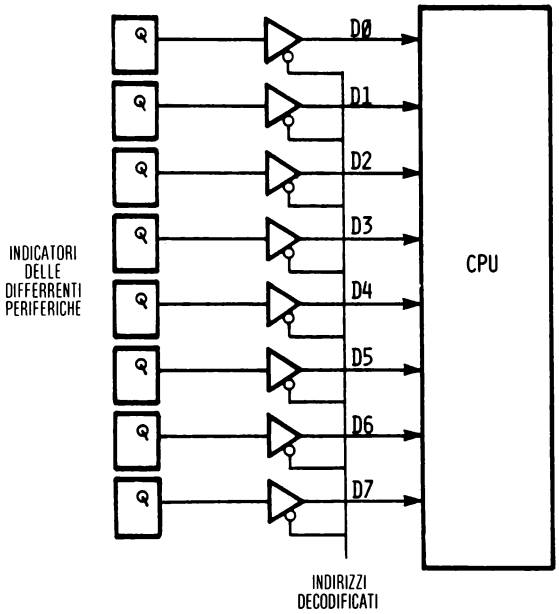


Fig. 3-29: Semaforo di stato di porta indicante modulo pronto

Il secondo metodo è quello di realizzare la codifica di priorità con una ROM di controllo o con un circuito codificatore di priorità. In tal modo la porta di stato contiene l'indirizzo del componente che in quel momento richiede servizio a più alto livello di priorità. Le figure 3-30 e 3-31 indicano il formato del byte e l'hardware necessario.

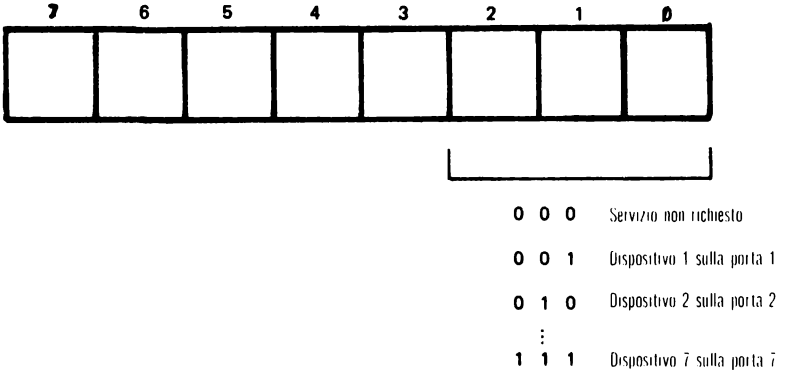


Fig. 3-30: Formato di Byte

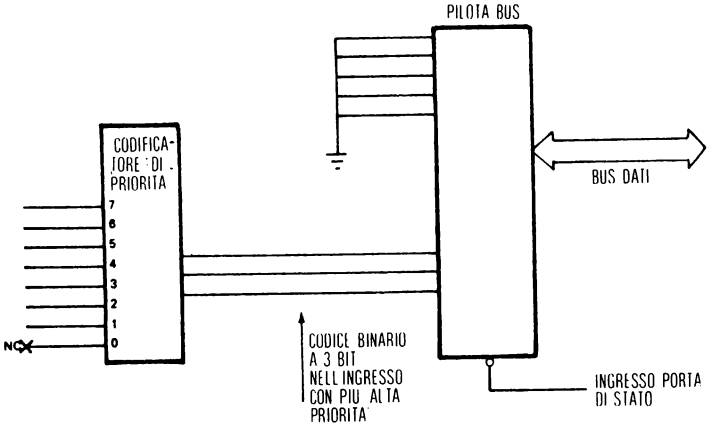


Fig. 3-31: Hardware del codificatore di priorità polling

Cambiando i cinque bit più alti di un qualsiasi altro codice, possono essere generati altri indirizzi di porte. Questo risparmia il controllo o la generazione dell'indirizzo di porta da parte della porta di stato del componente pronto *poichè la porta contiene l'indirizzo del componente pronto.* Il polling è la tecnica più comune e più semplice per il controllo di I/O. Esso non richiede hardware particolare e tutti i trasferimenti di ingresso e di uscita sono controllati da programma. I trasferimenti sono sincroni con l'esecuzione dei programmi.

Interruzioni

La tecnica di polling ha due limitazioni:

1. Essa consuma tempo di CPU poichè il processore controlla senza necessità e in continuazione lo stato di tutte le periferiche.
2. Essa è intrinsecamente lenta poichè essa controlla lo stato di tutti i componenti di I/O prima di ritornare ad uno specifico. Questo può essere inaccettabile in sistemi in tempo reale, nei quali una periferica richiede servizio entro un tempo specifico. In particolare, quando sono connesse al sistema periferiche veloci, il polling può non essere sufficientemente veloce per soddisfare le minime esigenze di servizio. Componenti veloci quali il disco «floppy» o il CRT richiedono un tempo di risposta quasi istantaneo per trasferire senza perdite.

Il polling è un meccanismo sincrono, nel quale i componenti sono serviti in sequenza. Le interruzioni sono invece un meccanismo asincrono. Il principio delle interruzioni è indicato in figura 3-22. Ciascun componente di I/O, o il suo controllore, è connesso ad una via di interruzione. Questa via controlla la produzione di una interruzione al microprocessore. Ogni qual volta uno dei componenti di I/O richiede servizio, esso genera un impulso o un livello di interruzione su tale via per richiedere l'attenzione del microprocessore.

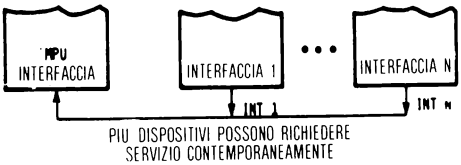


Fig. 3-32: Sequenza di interruzione

Un microprocessore controlla l'interruzione alla fine di ogni istruzione. Se essa è presente, il microprocessore serve la interruzione, altrimenti passa all'esecuzione della istruzione successiva. Questo è illustrato in figura 3-33. Durante l'esecuzione di alcuni processi critici, deve essere garantito che il programma non sia disturbato da interruzioni esterne. Un esempio di tale tipo è l'esecuzione della routine di caduta di potenza. La caduta di potenza può essere facilmente rivelata. Se il sistema è equipaggiato con una batteria di riserva («back-up») per la memoria, il processore può preservare i contenuti dei suoi registri in memoria e chiudere in modo ordinato l'intero sistema. Dopo che è stata rivelata la caduta di potenza, sono disponibili diversi millisecondi di tempo di elaborazione. È pertanto attivata una «routine» di caduta di potenza che sarà eseguita trascurando tutte le meno importanti richieste che possano presentarsi. Altre richieste sono mascherate. (La caduta di potenza è considerata una «interruzione non mascherabile»).

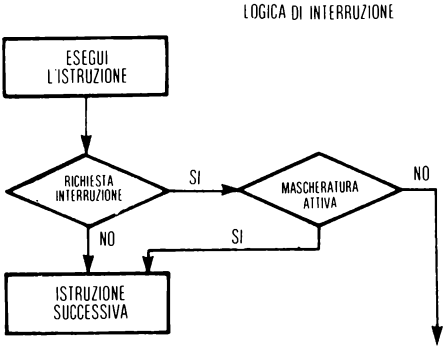


Fig. 3-33: Logica di interruzione

Lo scopo del «bit di mascheratura» (o registro quando sono presenti diversi livelli di interruzione) è quello di seguito indicato. Ogni qual volta tale bit è «on» è ignorata l'interruzione (vedi il diagramma di figura 3-33). Il servizio mascheratura è spesso chiamato «abilitazione». Un'interruzione è abilitata quando non è mascherata.

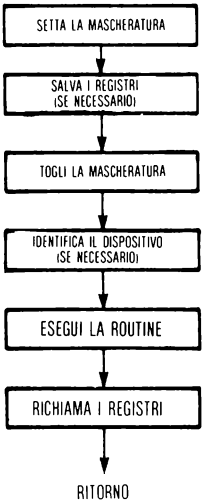


Fig. 3-34: Controllo di interruzione

Servizio di interruzione

Una volta che la richiesta di interruzione è stata ricevuta e accettata dal microprocessore, il componente deve essere servito. Per fare questo il microprocessore deve eseguire una specifica «routine» di servizio. Due diversi problemi si presentano. Anzitutto deve essere salvato lo stato del programma in esecuzione nel microprocessore al momento del servizio della interruzione.

Questo comporta che sia salvato il contenuto di tutti i registri del microprocessore, utilizzando lo *stack*. Al minimo deve essere scaricato sullo stack il contatore di programma («program counter») (PC), per poter caricare un nuovo indirizzo di diramazione nel PC e far girare il gestore di interruzione. Il salvataggio degli altri registri può essere fatto in hardware dal microprocessore o può essere realizzato sotto la responsabilità della routine di esecuzione dell'interruzione. Una volta che il PC è possibilmente gli altri registri sono stati salvati nello stack, il microprocessore procede secondo l'indirizzo del gestore dell'interruzione. A questo punto si presenta il secondo problema.

Piú componenti di I/O sono connessi alla stessa via di interruzione. Verso dove dovrà procedere il microprocessore per servire quel componente? Il problema è identificare il componente di I/O che ha prodotto la interruzione. Questa identificazione può essere fatta in hardware, in software, o mediante una combinazione dei due criteri. La ramificazione verso l'indirizzo del componente di I/O è chiamata *produzione di un vettore di interruzione*. Il metodo piú semplice, da un punto di vista hardware, non comporta la produzione di tale vettore. Una routine *software* può determinare la identità del componente che ha prodotto l'interruzione. È usato per questo il *polling*. La tecnica è illustrata in figura 3-36. La routine di identificazione dell'interruzione identifica ogni componente connesso al sistema.

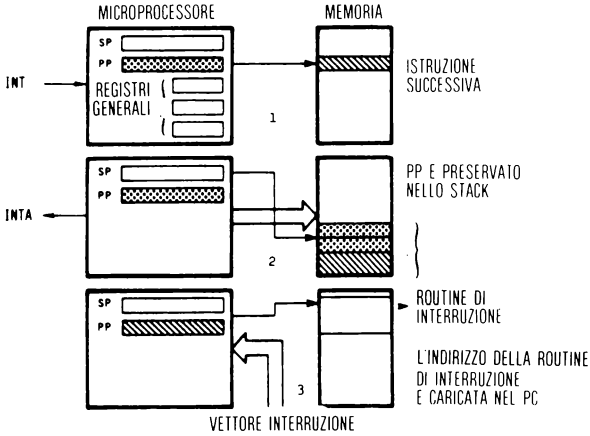


Fig. 3-35: Le tre fasi

Essa controlla il registro di stato, piú comunemente esaminando il bit 7. La presenza di un 1 in un dato bit indica che il componente ha prodotto una interruzione. Una volta identificato il componente, il servizio passa all'opportuno indirizzo della routine di gestione dell'interruzione. L'ordine secondo il quale è realizzato il polling determina quale componente è servito prima. Quella realizzata è una *priorità software*, nel caso piú componenti producano un'interruzione contemporaneamente.

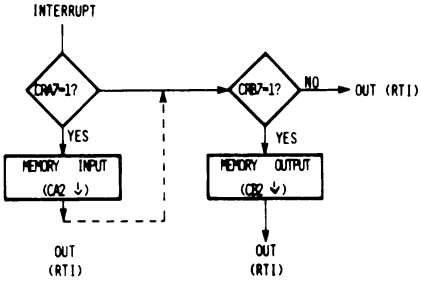


Fig. 3-36: Polling delle interruzioni

Un secondo metodo, pilotato dal software, ma con l'ausilio di un certo hardware addizionale, è apprezzabilmente piú veloce. Esso usa una lista concatenata (*daisy chain*) per identificare il componente che ha prodotto la interruzione. La figura 3-37 indica tale tecnica. Una volta preservati i registri, il microprocessore genera una indicazione di acquisizione dell'interruzione, che è indirizzata al componente n. 1. Se esso ha generato la interruzione pone il proprio indirizzo di identificazione nel bus dei dati, e tale indirizzo sarà riconosciuto dal microprocessore. In caso contrario l'indicazione di acquisizione è fatta propagare verso il componente n. 2. La stessa procedura è seguita qui e negli altri componenti. A causa dell'organizzazione fisica dei componenti, questo meccanismo è chiamato «daisy chain», ed è usato da molti PIO.

Il metodo piú veloce è chiamato *interruzione mediante vettori*. È responsabilità del componente di I/O produrre sia il vettore che presentare una propria *identificazione*, e ancora meglio produrre direttamente l'indirizzo della routine di gestione dell'interruzione. Se già il circuito di controllo fornisce la identità del componente, è un compito semplice per il software controllare una tabella che contiene l'indirizzo di ramificazione per ciascun componente. Questo è semplice dal punto di vista hardware ma non presenta le piú elevate caratteristiche di efficienza. Quest'ultima è ottenuta quando il microprocessore riceve una interruzione e l'indirizzo di ramificazione a 16 bit è diretto.

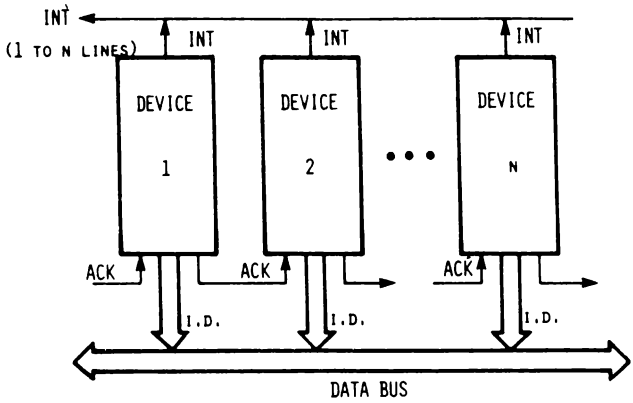


Fig. 3-37: Schema di concatenamento a margherita («daisy chain»)

In tal caso il microprocessore può direttamente procedere, passando alla posizione di memoria richiesta, iniziando il servizio della periferica. I nuovi PIC (Priority-interrupt-controller/controllore di interruzioni e priorità) realizzano tale tecnica.

Priorità

Un nuovo problema si presenta: diverse interruzioni possono essere generate nello stesso tempo. Il microprocessore deve pertanto decidere in quale ordine deve servirle. Una priorità è per questo associata a ciascun componente e il microprocessore servirà secondo l'ordine di priorità. Nel linguaggio dei calcolatori la priorità 0 è, per convenzione, quella piú alta, la priorità uno la successiva, e così via.

Tipicamente a priorità zero sarà la caduta di potenza (PFR o Power Failure Restart), il livello uno sarà usato dal CRT. Il livello 2 sarà lasciato libero per un eventuale secondo CRT. Al livello tre ci sarà il disco. Al livello cinque ci sarà la stampante. A livello 6 la teletype.

Il livello 7 sarà per commutatori esterni. In questo esempio non è usato il livello 4. Le priorità possono essere realizzate in hardware o in software. L'imposizione software delle priorità è stata già descritta. La routine di controllo dei componenti servirà quello a piú alta priorità. La implementazione della priorità hardware è anche possibile, così come è realizzata in recenti PIC. Mediante tale metodo è realizzata una mascheratura integrale di 8 bit che permette al programmatore di mascherare selettivamente qualsiasi livello di interruzione. La struttura fondamentale della logica di tali PIC è indicata in figura 3-38. Non è indicata la vettorizzazione degli indirizzi, ma semplicemente la generazione del vettore di livello. Un tale componente accetta 8 livelli di interruzioni. Essi compaiono nella parte destra della figura e porranno a 1 bit nel registro delle interruzioni.

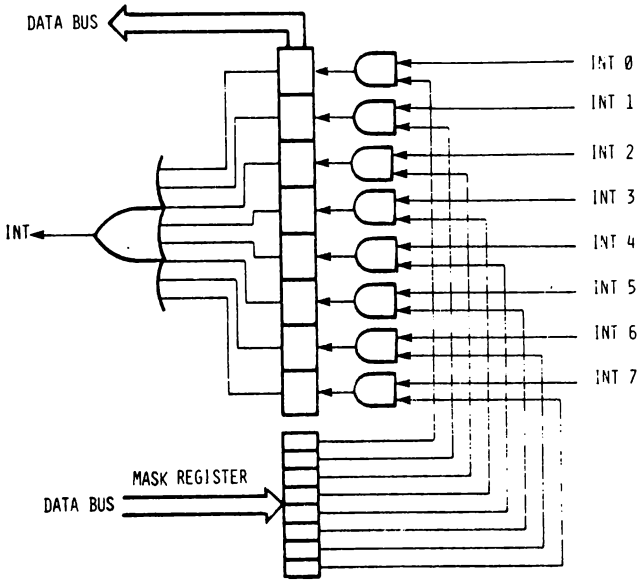


Fig. 3-38: Logica PIC

Il registro di mascheratura è usato dal programmatore per mascherare selettivamente livelli di interruzione. Tipicamente i livelli non utilizzati saranno mascherati. È tuttavia possibile mascherare livelli secondo criteri del programma. Una semplice porta AND permette la propagazione di una interruzione non mascherata. Il livello di interruzione di piú alta priorità è convertito in un codice di tre bit da un codificatore otto a tre. Una ulteriore operazione è eseguita: il livello della interruzione è confrontato con il contenuto del registro di priorità a tre bit. Il registro di priorità è posto ad 1 dall'utente. Questo eviterà la interruzione per un livello maggiore di n, dove n è la priorità abilitata. Si realizza cioè una mascheratura globale per ogni interruzione di livello superiore ad n. Un comparatore del PIC determina che il livello della interruzione è accettabile e genererà una richiesta finale di interruzione. Il microprocessore ha disponibile il vettore di interruzione a tre bit. Un PIC ancora piú sofisticato farà ancora di piú. PIC piú recenti infatti presentano una ramificazione diretta su un indirizzo a 16 bit. Questo è realizzato includendo una RAM di 8 x 16 bit registri entro il PIC. Il vettore di livello a tre bit è pertanto utilizzato per selezionare il contenuto di uno degli otto registri. Il contenuto di tali registri è caricato nel bus dei dati del microprocessore, e talvolta nel suo bus degli indirizzi. Ciò causa una automatica ramificazione all'indirizzo specificato. Naturalmente, tali registri sono caricati dal programmatore. In figura 3-39 è indicata la struttura di un PIC recente .

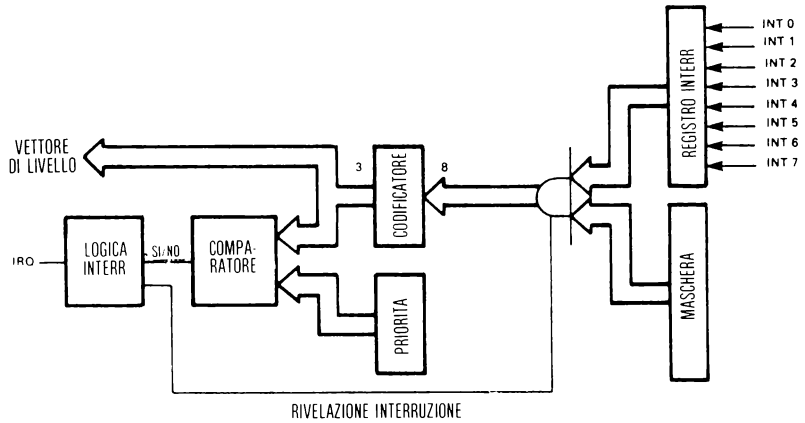


Fig. 3-39: Controllore di interruzioni con priorità

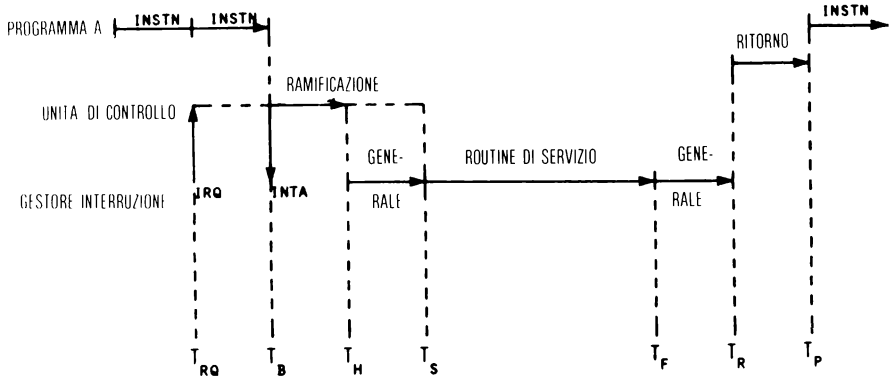


Fig. 3-40: Sequenza di interruzione

La figura 3-40 indica la sequenza degli eventi durante una interruzione. Procedendo in figura da sinistra verso destra, il programma A è in esecuzione finchè non e generata una richiesta di interruzione al tempo T_{RQ}. Si terrà conto della interruzione alla fine della istruzione, al tempo T_B. L'unità di controllo del microprocessore realizza pertanto la ramificazione all'indirizzo necessario. Una volta che tale ramificazione è eseguita, il gestore della interruzione (la terza linea della figura 3-40) inizia la esecuzione. Esso dovrà spendere un certo tempo di overhead per salvare i registri, che non sono stati salvati automaticamente dalla unità di controllo del microprocessore.

La routine di servizio del componente passa pertanto all'esecuzione. Alla fine dell'esecuzione, i registri devono essere ripristinati (tempi da T_F a T_R). È pertanto eseguita una routine di ritorno, e l'unità di controllo ripristina il contenuto precedente del contatore di programma (PC), prelevato dallo stack, in modo che possa proseguire l'esecuzione del precedente programma A. Il programma A riparte a T_P.

I tempi da T_{RQ} a T_S sono i tempi di risposta all'interruzione, cioè il tempo totale che intercorre tra la richiesta di interruzione e l'istante in cui la routine di servizio ha effettivamente inizio, facendo il lavoro richiesto. Alcuni costruttori ritengono che il tempo di risposta sia solo quello tra T_{RQ} e T_H. La lunghezza totale del tempo perduto dal programma è da T_B a T_P. L'overhead totale dell'interruzione è realmente da T_B a $T_S + T_F$ fino a T_R. Tali valori variano apprezzabilmente da un microprocessore all'altro.

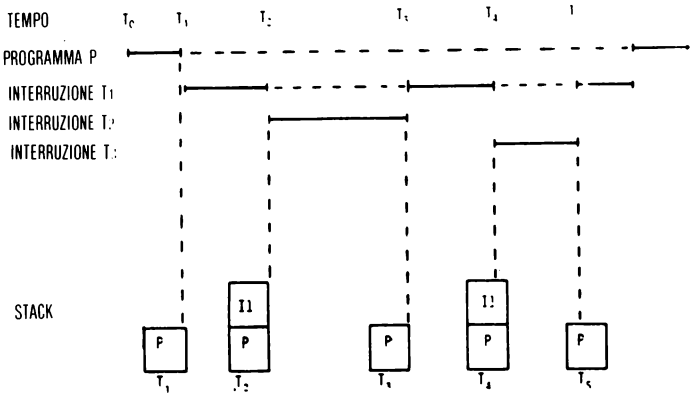


Fig. 3-41: Interruzioni durante uno «stack»

Interruzioni multiple e uso dello stack

La figura 3-41 indica il ruolo dello stack durante le interruzioni multiple. Al tempo T_O il programma è in esecuzione. Al tempo T_1 l'interruzione I_1 è accettata. Sono pertanto caricati nello stack i registri usati dal programma P (vedere la parte, più bassa dell'illustrazione, sulla sinistra). L'interruzione I_1 è eseguita fino a T_2. In tale istante è prodotta l'interruzione I_2, e si assume che essa sia a priorità più alta. L'interruzione I_1 è sospesa, come precedentemente il programma P.

I registri usati per I_1 sono caricati nello stack. Ciò è illustrato in figura 3-38 nella parte bassa dell'illustrazione, dal tempo T_2. È pertanto eseguita l'interruzione I_2. Questa è la terza linea della figura 3-41. L'interruzione I_2 è eseguita fino al completamento, al tempo T_3. In quel momento il contenuto dello stack viene ricaricato nel

microprocessore e soltanto P è lasciato nello stack (vedi figura 3-41: lo stack contiene soltanto P all'istante T_3).

L'interruzione I_1 ritorna in esecuzione e, al tempo T_4, è nuovamente sospeso da un'altra interruzione, I_3, di più alta priorità. Nuovamente due livelli sono nello stack: I_1 e P al tempo T_4 (vedi fig. 3-41).

L'interruzione I_3 è eseguita fino al completamento, al tempo T_5. Ora I_1 è ricaricata dallo stack e riprende la sua esecuzione. Questa volta essa gira fino al completamento, al tempo T_6, quando il programma P è prelevato dallo stack e riprende la sua esecuzione. È da osservare che il numero di livelli presenti nello stack è eguale al numero dei programmi sospesi, cioè, al numero di linee orizzontali tratteggiate presenti in ogni intervallo. L'esempio descritto spiega l'uso dello stack durante le interruzioni multiple. Naturalmente, se si possono verificare simultaneamente un numero maggiore di interruzioni, è necessario disporre di uno stack più grosso per contenere il numero di livelli possibili.

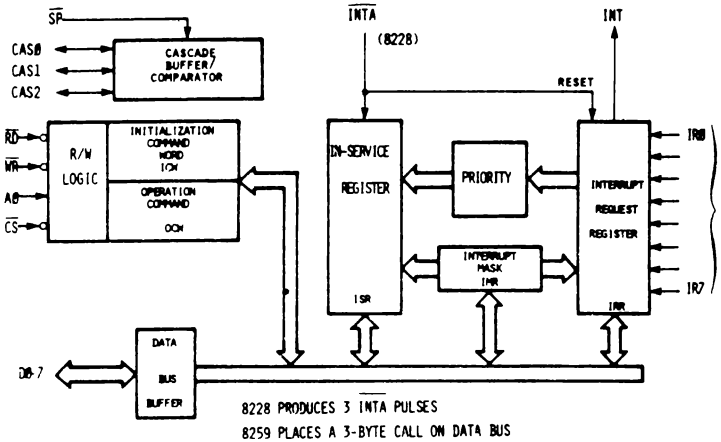


Fig. 3-42: Controllore di interruzione 8259

Accesso Diretto in Memoria (DMA)

Le interruzioni garantiscono la risposta più veloce possibile ai componenti di ingresso/uscita. Tuttavia il servizio del componente è realizzato in software. Esso può essere non sufficientemente veloce da servire processi che richiedono trasferimenti rapidi da memoria come un disco o uno schermo CRT. Nuovamente la soluzione è sostituire al software l'hardware. La routine di software che realizza il trasferimento tra la memoria e il componente è rimpiazzato da un processore hardware specializzato, il DMAC o Direct-Memory-Access Controller (Controllore di accesso diretto in memoria). Un DMAC è un processore specializzato per realizzare trasferimenti di dati ad alta velocità tra la memoria e il componente. Per realizzare tali trasferimenti il DMAC richiede l'uso di entrambi i bus dei dati e degli indirizzi.

Le filosofie dei DMAC differiscono nel modo nel quale essi ottengono accesso ai bus. Per esempio un DMAC può, durante la sua attività, tenere sospeso il processore centrale, può fermarlo, o può utilizzare suoi cicli, o ancora può allungare intervalli del segnale di temporizzazione.

Alcuni DMA, come il DMA a ripristino dinamico della memoria, possono usare alcune porzioni del ciclo di memoria, quando essi «sanno» che il processore non richiede l'uso del bus dei dati e di indirizzo. Una discussione dettagliata delle filosofie DMA è fuori dallo scopo di questo libro. L'approccio più semplice, e quello più comunemente usato dai microprocessori, è di sospendere il funzionamento del microprocessore. Questa è la ragione per la quale sono usati bus di tipo tri-state per i dati e gli indirizzi. L'organizzazione di un sistema DMA è indicata in figura 3-43. Ciascun componente spedirà il proprio segnale di interruzione al DMAC e non al microprocessore. Quando il DMAC riceve una interruzione dal componente genera un segnale speciale per il microprocessore, il segnale HOLD. Esso sospende il microprocessore, ponendolo in uno stato dormiente.

Il microprocessore completa la sua istruzione e lascia il bus dei dati e di indirizzi in uno stato di alta impedenza. Si suol dire «far fluttuare» i bus. A tal punto esso «dorme» e risponde con una conferma all'HOLD. Alla ricezione della conferma all'HOLD, il DMA sa che i bus sono stati liberati. Porrà pertanto automaticamente un indirizzo nel relativo bus, che specifica l'indirizzo di memoria dove il trasferimento di dati deve aver luogo. Un DMAC connesso a 8 componenti di I/O conterrà 8 registri di indirizzo di 16 bit per lo scopo. Naturalmente il contenuto di questi indirizzi è stato specificato dal programmatore per ciascun componente. Il DMAC specifica l'indirizzo nel quale il trasferimento deve essere fatto, pertanto genera un segnale «lettura» o «scrittura», e lascia che il componente riceva o trasmetta

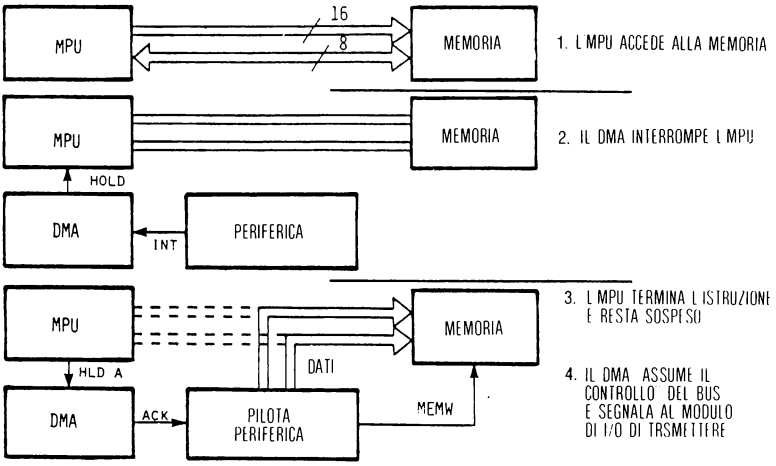


Fig. 3-43: Criteri operativi del controllore DMA

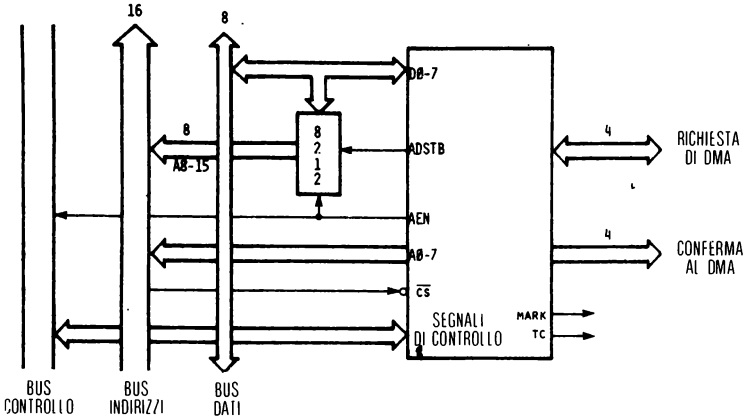


Fig. 3-44: DMAC Intel

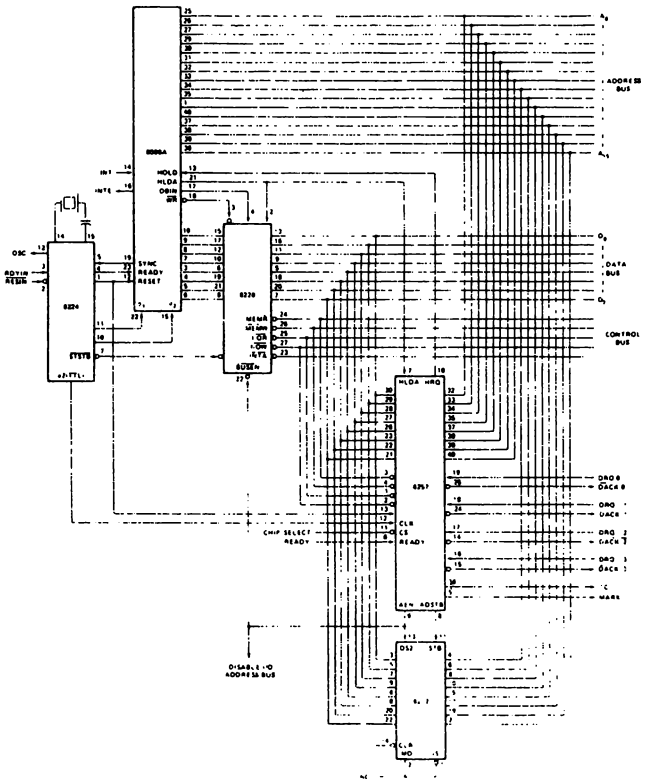


Fig. 3-45: Piastra di CPU in DMA

i dati attraverso il bus dei dati. In aggiunta un DMAC contiene un circuito controllore di sequenza del trasferimento dei blocchi. Questo è particolarmente importante per trasmettere blocchi di dati (nel caso di dischi) o sequenze di dati (nel caso di CRT). Il DMAC dispone di un registro contatore per ciascun componente. Tipicamente è realizzato un contatore ad 8 bit per permettere il trasferimento da 1 a 256 parole. Dopo ogni trasferimento di parola il contatore è decrementato. Il trasferimento si ferma quando il contatore va a 0 o quando scompare la richiesta di DMA da parte del componente.

Il vantaggio del DMA è di garantire il trasferimento dal componente alla velocità più alta possibile. Il suo svantaggio è, naturalmente, di rallentare il funzionamento del processore. Il DMA è un componente molto sofisticato, allo stesso livello del microprocessore. Esso è inoltre costoso poichè non è venduto in quantità pari a quelle di un microprocessore. In molti casi è più economico dedicare un microprocessore con memoria per realizzare trasferimenti a blocchi, che usare circuiti DMA. Ad esempio, in figura 3-44 è indicata la struttura di un DMAC della INTEL, e in figura 3-46 quella del DMAC del MOTOROLA 6800. Il controllore DMA indicato in figura 3-46 è un controllore a sottrazione di ciclo. Il bus di indirizzo e il segnale R/W fluttua fino a 500 ms. Tuttavia la durata massima della sospensione non può superare 5 microsecondi, poichè i registri dinamici del 6800 perdono il loro contenuto dopo tale tempo. Il nuovo DMAC 6844 della Motorola può

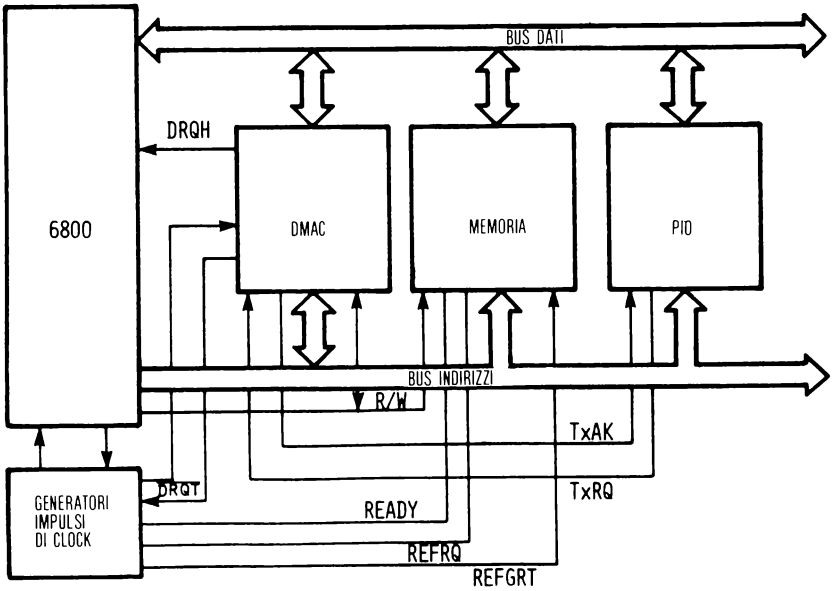


Fig. 3-46: Connessioni del DMAC della Motorola

operare in tre modi: alt - blocchi, alt-furto di ciclo (1 trasferimento di byte), e furto di TSC. In «alt-blocchi», una richiesta di trasferimento in T x RQ ferma il 6800 e l'indicazione di 0 nel conteggio dei byte lo fa ripartire. Questo è un trasferimento a blocchi. In alt-furto di ciclo è trasferito un solo byte. Esso ha quattro canali DMA con 16 bit di indirizzo e un contatore a 16 bit.

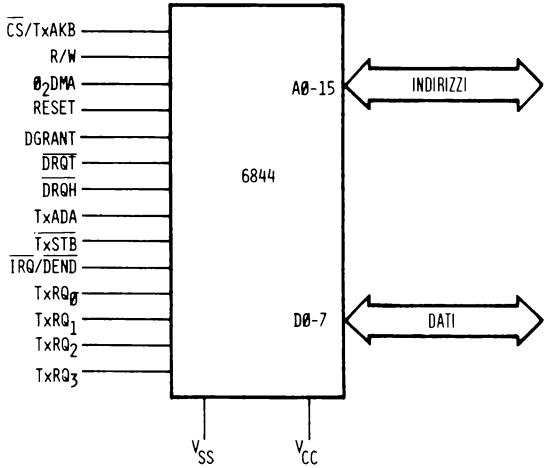


Fig. 3-47: Il modulo DMAC

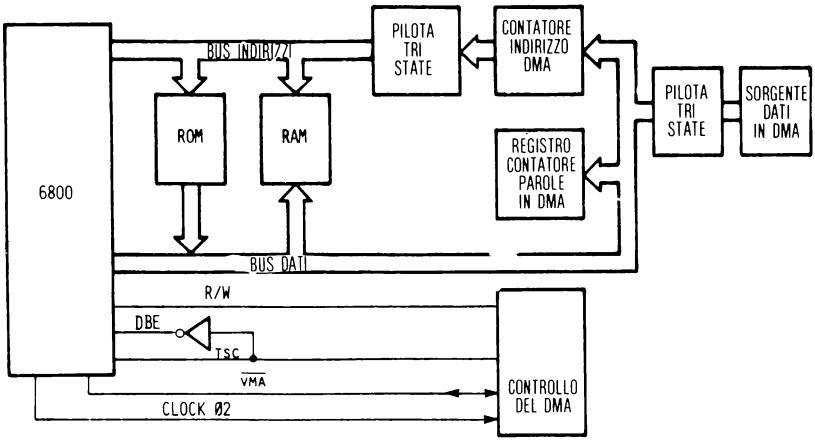


Fig. 3-48: Criteri di operazione

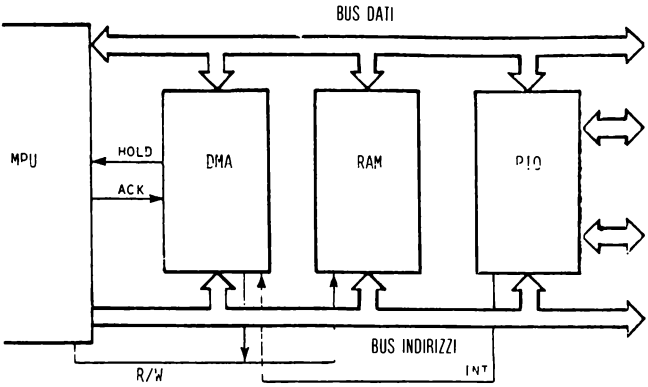


Fig. 3-49: Diagramma a blocchi del DMA

La massima velocità di trasferimento è di un megabyte al secondo. La struttura del sistema è indicata in figura 3-49 e 3-50. Lo 8257 della Intel dispone di quattro canali e funziona con una semplice sospensione dello 8080 (per qualsiasi intervallo di tempo). Esso richiede un controllore di trasferimento (latch) 8212 per i bit da 8 a 15 del bus di indirizzi. Le figure 3-45 e 3-50 indicano gli schemi circuitali. Infine l'interconnessione dell'Am 9517 della AMD a un 8080 opera con uno schema indicato in figura 3-51.

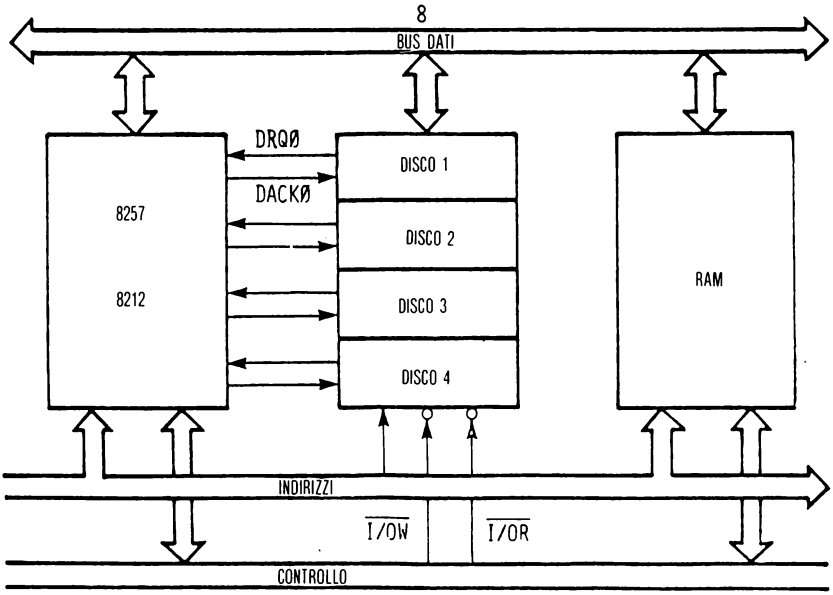


Fig. 3-50: 4 canali dell'8257

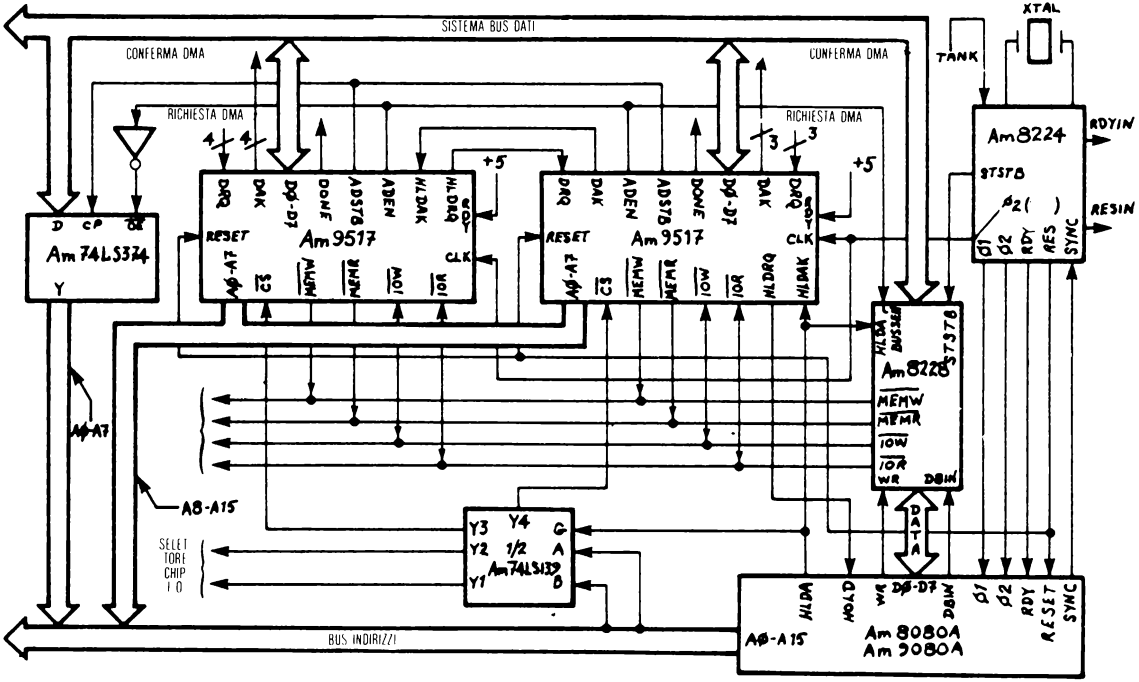


Fig. 3-51: Esempio di applicazione dell'AM9517

CIRCUITI UTILI DI DIVERSO TIPO

Oltre agli elementi del microprocessore, il circuito integrato microprocessore, la RAM, la ROM e gli I/O, sono necessari elementi in logica random di diverso tipo per completare la configurazione del sistema. Essi sono porte logiche, invertitori, monostabili, multiplatori, contatori e trigger di Schmidt.

I componenti AND, OR, NAND, NOR non saranno trattati, poichè si suppone che il lettore abbia già familiarità con questi elementi logici (vedere il riferimento C 201 per una discussione sulle porte logiche).

Il primo elemento trattato è il *monostabile* (one-shot), o componente monostabile asincrono. Esso è in realtà un circuito analogico. Quando è applicato all'ingresso un impulso, il monostabile presenta un impulso di lunghezza variabile in uscita. La durata dell'impulso di uscita non dipende da quello applicato all'ingresso. Esso è determinato da due elementi di temporizzazione, normalmente una resistenza e una capacità. I monostabili sono pertanto utili quando deve essere allungata la durata di un impulso. Due esempi sono: impulsi di reset o impulsi di interruzione. Il fatto che esso sia un elemento analogico implica che presenti una affidabilità minore degli altri elementi del circuito. Infatti esso è per natura più sensibile al rumore dell'alimentazione ed ha problemi di disaccoppiamento. È considerato un buon criterio di progetto evitare quanto possibile l'uso del monostabile. Una sua struttura tipica è indicata in figura 3-52.

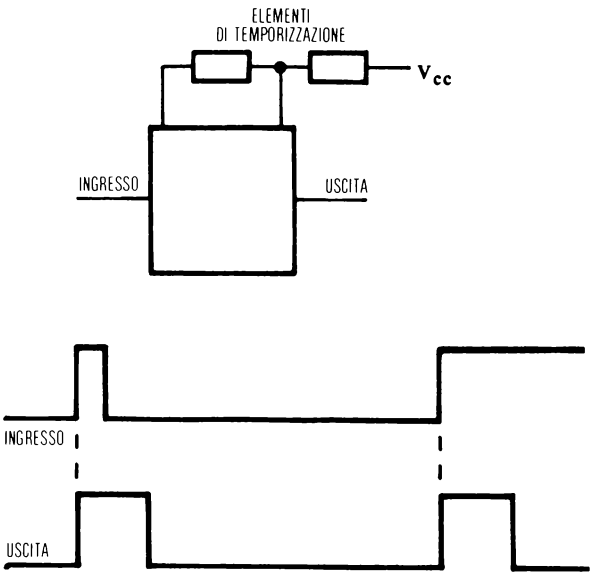


Fig. 3-52: Periodo di impulso di un monostabile

Multiplatori e demultiplatori: funzionano come commutatori digitali. Un multiplatore accetta un certo numero di ingressi e li multipla in una singola linea. Un demultiplatore separa tali segnali da una linea in più linee. Così operando, i multiplatori e i demultiplatori sono molto simili a commutatori rotanti, che possono essere controllati in modo digitale mediante indirizzi di ingresso. I multiplatori sono necessari per il progetto di circuiti integrati con ram dinamiche multiplate e di interfacce di ingresso scandite. I multiplatori sono spesso usati come decodificatori, oltre a svolgere la funzione complementare dei multiplatori. I criteri operativi dei multiplatori sono indicati in figura 3-53. Il *trigger di Schmidt* è un elemento di interfaccia per rendere immuni al rumore segnali TTL e convertirli in segnali TTL con un unico fronte di commutazione. Esso realizza tale funzione attraverso una *i-*

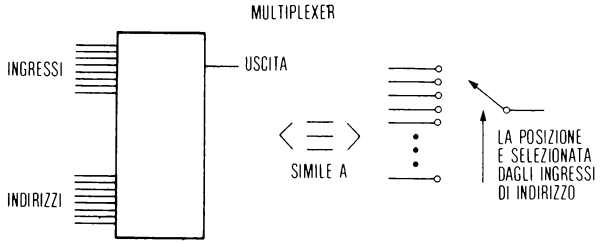


Fig. 3-53: Modo di operare di un multiplatore

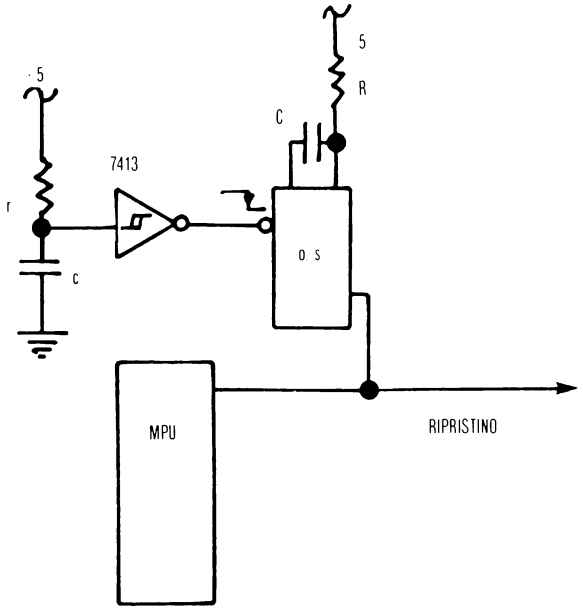


Fig. 3-54: Circuito di reset di MPU

steresi di ingresso. Il segnale di ingresso deve passare attraverso due soglie prima che l'uscita possa cambiare.

È così generata una transizione pulita da un segnale di ingresso rumoroso o con variazione di livello lenta. In figura 3-54 è indicata una tipica applicazione di uno Schmidt trigger che usa 7413. Usato con un monostabile per la generazione di un reset pulito, il trigger di Schmidt farà scattare il monostabile all'accensione dell'alimentatore. Il monostabile genererà un impulso di reset della durata minima, pari al numero minimo di impulsi del segnale di temporizzazione richiesti dal particolare microprocessore.

In molti casi, al reset, è richiesto un differente indirizzo di inizializzazione. Usando un multiplatore, come in figura 3-55, nuove vie di indirizzo possono essere generate, come richiesto. Nell'applicazione indicata, l'indirizzo multiplato è commutato alla nuova via di indirizzo ogni qualvolta capiti un reset. Per ritornare alle vecchie vie di indirizzi, il flip-flop è resettato a 0.

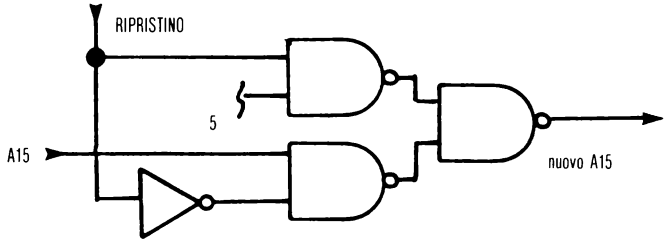


Fig. 3-55: Vettore di indirizzo per reset convertito in un differente vettore reset

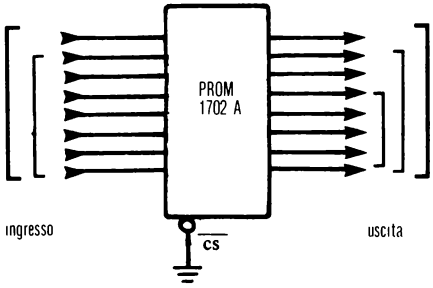


Fig. 3-56: Conversione di codice usando una PROM

Saranno pertanto scelte le vecchie vie di indirizzo quando il programma di reset ha completata la sua esecuzione.

Oltre a questi componenti, che sono il filo e l'ago di un sistema, esistono altre applicazioni delle ROM, oltre a quella di memorizzare programmi. In figura 3-56 è indicata un'applicazione di una ROM standard come convertitore di codice. Inseren

do questa ROM nella via dei dati in parallelo tra il componente di ingresso e il microprocessore o tra il microprocessore ed il componente di uscita, si realizza la conversione da codice ASCII a EBCDIC. Sono possibili altri tipi di conversione. Altro uso è come controllare passo passo («watchdog») del software. Una sequenza campione, prodotta usando un analizzatore logico, è generata dalla ROM. Questa sequenza statica è confrontata con quella prodotta dallo stesso analizzatore logico.

Se è presente per caso un cambiamento, ciò indica un malfunzionamento software. Gli elementi di questo circuito di controllo passo passo sono indicati in figura 3-57. Una volta che un guasto software è stato rivelato l'hardware interrompe il microprocessore. Tale interruzione gli indicherà che si è verificato un guasto software. Potranno essere fatti girare a questo punto programmi di autodiagnosi per determinare la causa del problema.

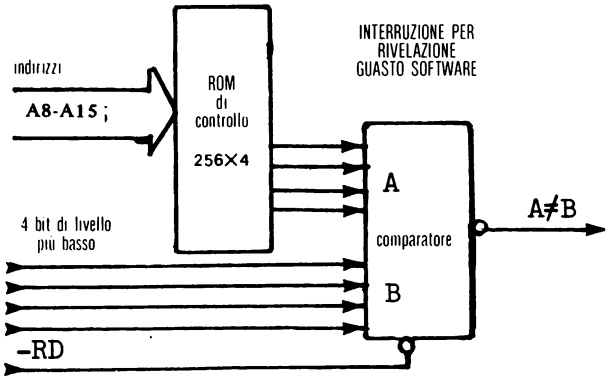


Fig. 3-57: ROM di rivelazione di guasto software mediante tabella ottenuta da un analizzatore logico

CONCLUSIONI

In questo capitolo sono state descritte le tecniche fondamentali di ingresso-uscita ed i relativi componenti. In un sistema reale il sistemista sceglierà la combinazione degli algoritmi hardware e software necessari per soddisfare le sue esigenze e limitazioni di costo. Nuovi circuiti integrati saranno presentati in futuro. Essi offriranno ancor maggiore efficienza per la gestione del trasferimento in ingresso/uscita ad alta velocità.

Prima di terminare la trattazione delle interfacce di CPU, ed i problemi di I/O, occorre considerare alcuni semplici circuiti necessari per assiemare il sistema. Nelle figure da 3-52 a 3-57 sono indicati alcuni circuiti utili. Essi includono monostabili, un circuito di reset e convertitori di codice. È di seguito fatta una loro descrizione.

Il successivo problema, molto importante, da risolvere è l'interfacciamento delle periferiche. Quest'argomento sarà sviluppato nel capitolo 4.

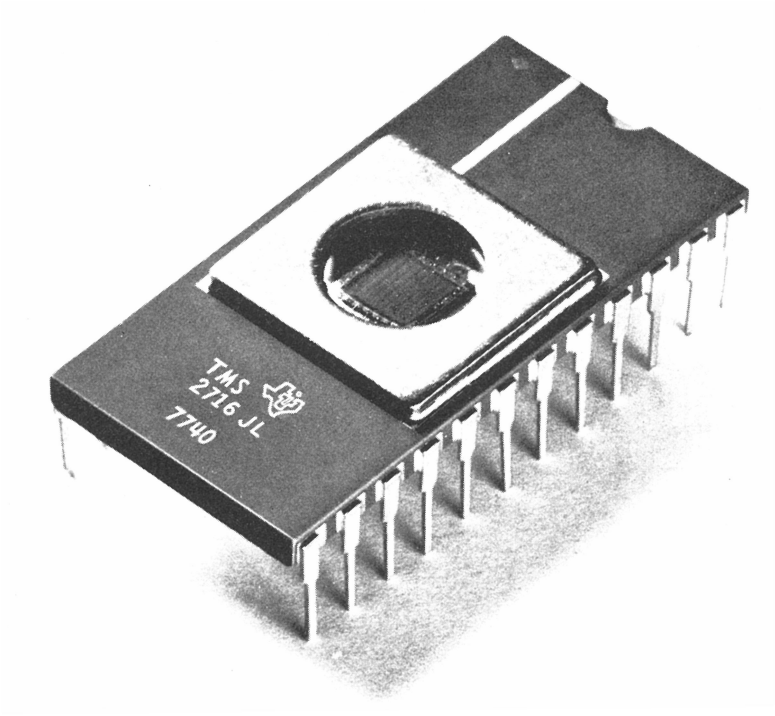


Fig. 3-58: EPROM 2716 della Texas Instruments

CAPITOLO 4

INTERFACCE VERSO PERIFERICHE

INTRODUZIONE

Adesso che la CPU, la memoria e i moduli di ingresso/uscita sono connessi e in funzione, in che modo connettere la teletype? Cosa fare per il nastro perforato, la tastiera e la linea telefonica? Esse sono tutte *periferiche* che permettono all'utente o ad un altro calcolatore di comunicare con il sistema. In questo capitolo sarà indicato come interfacciare le più comuni periferiche, cioè:

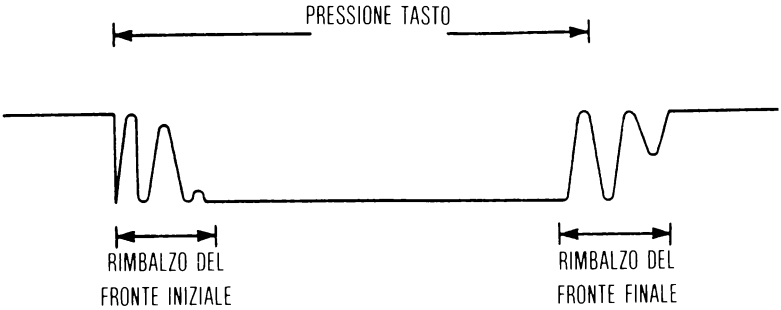
- tastiere (incluse le tastiere in codice ASCII)
- schermi LED
- teletype (TTY)
- lettori di nastro perforato (PTR)
- motori passo passo
- lettori di striscia magnetica di carte di credito
- interfaccia Tarbell
- registratori a cassette
- schermi CRT
- dischi Floppy
- sintetizzatori musicali
- interfaccia di RAM dinamica

TASTIERE

Una tastiera consiste di *commutatori attivati da pressione o dal tocco organizzati a matrice.* Rilevare quale tasto è stato premuto richiede normalmente la combinazione di un supporto hardware e/o software. Sono disponibili due tipi fondamentali di tastiere: *codificate e non codificate.* Tastiere codificate includono l'hardware necessario per rivelare quale tasto è stato premuto e per memorizzare il dato finché avviene un'altra battuta. Tastiere non codificate non hanno hardware e devono essere analizzate da una routine software o da hardware speciale.

Il rimbalzo

Uno dei problemi più comuni per il singolo commutatore è il *rimbalzo.* Il rimbalzo del tasto avviene per il fatto che quando sono chiusi i contatti di un commutatore meccanico, essi oscillano per un breve intervallo di tempo, prima di stabilizzare



- IL RIMBALZO DURA 10-20 ms.
- SOLUZIONE HARDWARE: FILTRO R/C
- SOLUZIONE SOFTWARE: VERIFICA DELLO STATO DEL TASTO PER 20 ms.

Fig. 4-0: Rimbalzo di tasto

un contatto sicuro. La stessa cosa avviene quando il contatto è aperto. In figura 4-0 è indicato un diagramma tempo/resistenza di un tipico commutatore.

La soluzione è aspettare che lo stato del tasto rimanga stabile per circa 20 millisecondi. Ciò può essere ottenuto attraverso un *filtraggio hardware* o attraverso una *routine di ritardo* software. Il circuito hardware è indicato in figura 4-1 e deve essere ripetuto per ogni tasto. Tale circuito è utile per i commutatori del pannello frontale di un sistema. Per un gran numero di tasti è spesso usato il software.

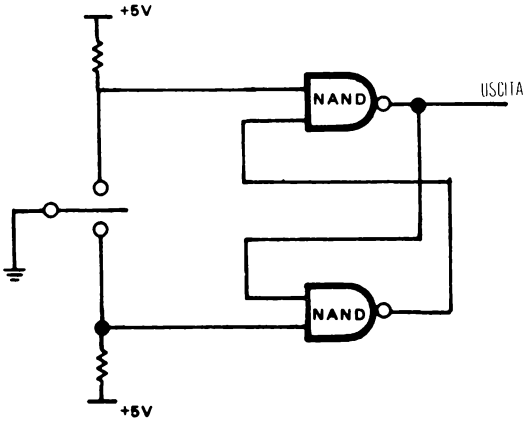


Fig. 4-1: Circuito di smorzamento

Tastiere non codificate

Normalmente la tastiera è organizzata a righe e colonne, con una matrice di n x m tasti. Si può scandire un gruppo di linee in sequenza progressiva e controllare l'eccitazione delle altre linee per rivelare la coincidenza di eccitazione (vedi fig. 4-3). Questa identificazione del tasto è conosciuta come «scansione per righe». Una volta rivelata la coincidenza, essa è controllata per 20 millisecondi o più, per verificare la sua stabilità. Solo allora sono generati i dati corrispondenti.

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

Fig. 4-2: Una tastiera a 16 tasti

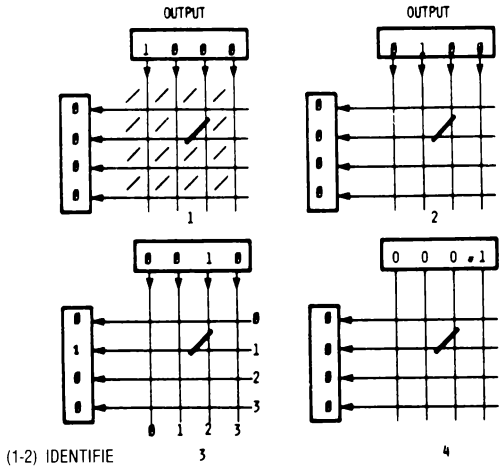


Fig. 4-3: Decodifica di tastiera di tipo a sequenza progressiva

Tastiere più grosse richiedono più linee di selezione e di controllo. La fig. 4-4 mostra come un decodificatore da 4 a 16 vie è adatto per una matrice di 64 tasti, con 4 bit di uscita e 4 bit di ingresso dalle porte di I/O del microprocessore.

La figura 4-5 mostra una semplice matrice a 12 tasti che usa 4 bit di uscita e tre di ingresso di un sistema microprocessore F8.

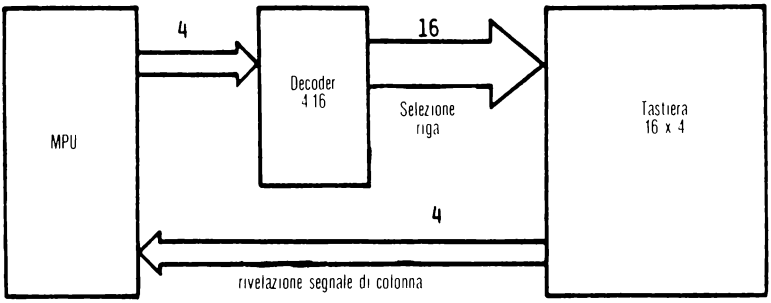


Fig. 4-4: Decodificatore da 4 a 16 vie con tastiera

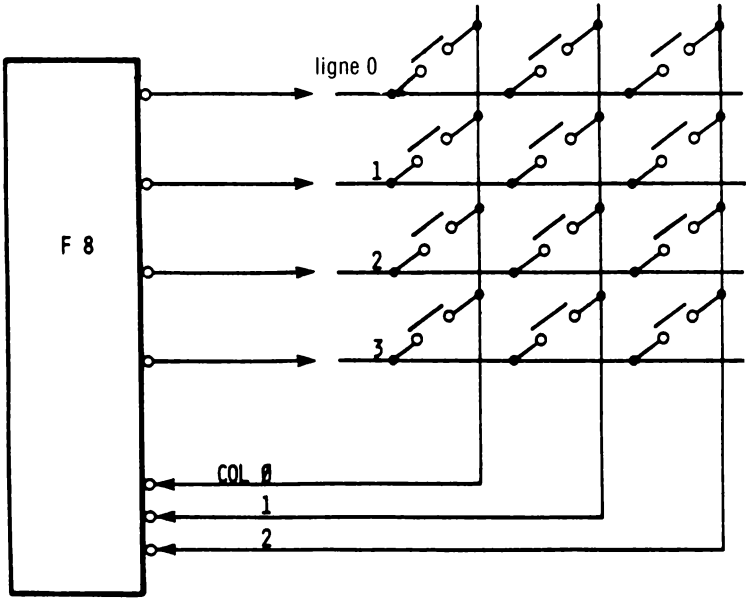


Fig. 4-5: Matrice F-8 a 12 tasti

«Rollover»

Esso è il problema che si verifica quando è premuto più di un tasto contemporaneamente. È importante rivelare tale situazione per evitare che siano generati codici

errati. Le tre tecniche principali usate per risolvere questo problema sono il *rollover di due tasti, il rollover di n tasti, e l'esclusione di n tasti.*

Il **rollover a due tasti** protegge il caso in cui due tasti sono premuti contemporaneamente. Sono usate due filosofie. La più semplice delle due ignora l'indicazione da tastiera finché è rivelata la chiusura dei contatti di un solo tasto. L'ultimo tasto che rimane premuto è quello corretto. Questa filosofia è quella normalmente usata quando sono utilizzate routine software per scandire e decodificare la tastiera. La seconda filosofia è più comunemente usata da componenti hardware. L'impulso di campionamento relativo alla chiusura di un secondo tasto è bloccata finché quello relativo al primo sia presente. Ciò è realizzato mediante un meccanismo interno di ritardo che dura finché il primo tasto è premuto. Chiaramente, per una più efficiente protezione, il rollover può essere controllato per più di due tasti.

Il **rollover a n tasti** può ignorare tutti i tasti premuti finché lo sia uno solo, o può memorizzare l'informazione in una memoria temporanea interna. Un costo significativo della protezione del rollover a n tasti consiste nella presenza in gran parte dei sistemi di un diodo in serie in ciascun tasto per eliminare il problema creato quando sono premuti tre tasti adiacenti ad angolo retto («tasto fantasma»). Ciò aumenta il costo in modo significativo ed è raramente usato in sistemi a basso costo.

La esclusione di n tasti considera premuto solo un tasto. Qualunque altro tasto che possa essere premuto e rilasciato non genera alcun codice. Per convenzione può essere il primo tasto premuto a generare il codice, o l'ultimo tasto lasciato premuto. Il sistema è più semplice da implementare ed è molto spesso usato. Tuttavia è poco accettabile per l'utente perché riduce la velocità di stampa: ciascun tasto deve essere completamente rilasciato prima che il successivo sia premuto.

Tecnica di inversione di via

La tecnica fondamentale usata per identificare il tasto che è stato premuto nella tastiera è la scansione per riga, già descritta. Tuttavia, per la disponibilità del circuito di interfaccia universale parallelo, il PIO, può ora essere usato un altro metodo. Esso è la tecnica di inversione di via. Questo metodo usa una porta completa di un PIO, ma sarebbe più efficiente se realizzato in software (più veloce).

Questo metodo è illustrato sotto, nell'esempio di una tastiera a sedici tasti. Una porta del PIO è dedicata all'interfaccia della tastiera. L'identificazione della chiave è realizzata essenzialmente mediante quattro istruzioni soltanto. In pratica, può essere necessario un numero maggiore di istruzioni a causa della specifica struttura del PIO usato.

Primo punto: l'uscita

Inizialmente, le otto vie del PIO sono configurate come quattro vie di ingresso e quattro vie di uscita.

Ciò sarà realizzato caricando un'opportuna trama di dati nel registro di direzione, che controlla la direzione delle vie. Nell'esempio il registro di direzione è carica-

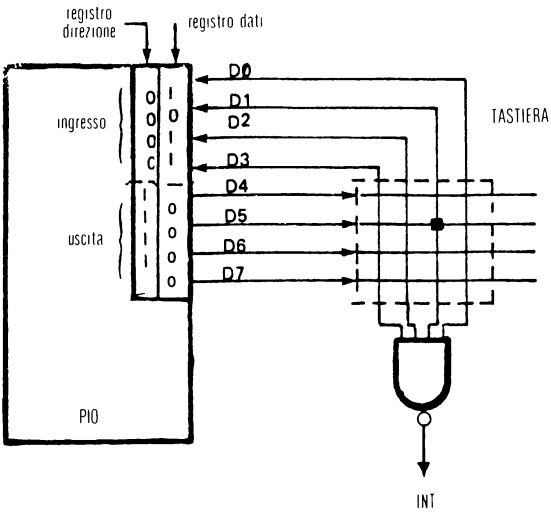


Fig. 4-6: Inversione di linea: primo passo

to con il valore «00001111». Ciò significa configurare le vie dei dati da D0 a D3 come ingressi, e le vie dei dati da D4 a D7 come uscite. Da D0 a D3 sono le uscite di riga della tastiera. Da D4 a D7 sono gli ingressi di colonna verso la tastiera. Si assume che il valore iniziale del registro dati siano tutti zeri. In altre parole, quattro 0 sono uscite sulle vie da D4 a D7, cioé gli ingressi di riga verso la tastiera. Ogni qual volta un tasto è premuto nella tastiera, l'uscita normale nella colonna, che è uno, è portata a massa dalla chiusura del tasto. Come risultato un valore «zero» apparirà nell'uscita di colonna nella quale è stato premuto un tasto.

Nell'esempio indicato nell'illustrazione uno zero appare nella via D1 (la terza colonna da sinistra della tastiera). Le altre tre uscite di colonna, cioé le vie D0, D2, D3, non sono state portate a massa da nessuna chiusura di tasto, e presentano un'uscita «uno». La rivelazione della chiusura vera e propria del tasto può essere realizzata in due modi. Una porta NAND, indicata in figura sotto la tastiera, può essere usata per generare un'interruzione al microprocessore. Altra comune alternativa è l'uso di un programma polling che legge il contenuto del registro dati e rivela il fatto che uno zero è presente in qualsiasi delle vie da D0 a D3. Il problema ancora da risolvere è identificare quale tasto è stato premuto. L'informazione già disponibile nel registro dati, cioé 10110000, non è sufficiente. È identificata la colonna, ma non la riga. Questo problema è stato superato nella tecnica di scansione per riga presentando successivamente un «uno» in ciascuna riga. Qui sarà usato un metodo più elegante, che fornirà la stessa informazione in un numero minore di passi.

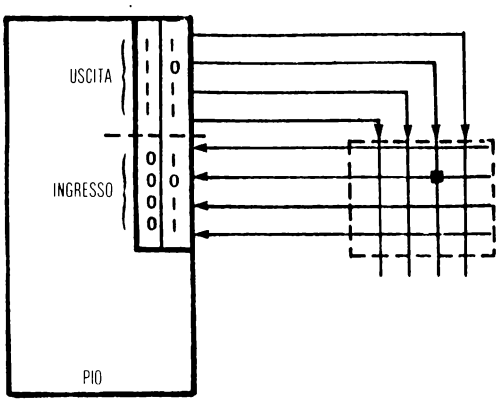


Fig. 4-7: Inversione di linea: secondo passo

Punto due: inversione di via

A questo punto la direzione delle 8 vie è semplicemente invertita. Gli ingressi diventano uscite, e viceversa. Questo è indicato sulla destra del disegno. Una semplice istruzione è necessaria per realizzare questa inversione di via: «complementa i contenuti del registro di direzione». Naturalmente questo comporta che questa istruzione sia disponibile. In alcuni microprocessori sono necessarie due o anche tre istruzioni per realizzare ciò all'esterno del microprocessore.

I contenuti del registro di direzione sono ora «11110000». Come risultato i contenuti del bit da D0 a D3, che erano precedentemente ingressi, ora sono uscite. Il valore «1011» è pertanto uscita delle colonne della tastiera. Come risultato, le vie da D4 a D7 sono condizionate dalle righe della tastiera. In questo esempio il valore risultante da D4 a D7 è «1011». Ogni qual volta un tasto è stato premuto, è generato uno «0» in uscita. Infine è sufficiente leggere i contenuti del registro dati per sapere quale tasto è stato premuto. I contenuti del registro dati sono nel nostro esempio «10111011». Ciò indica che il tasto è stato premuto all'intersezione della terza colonna e della terza riga. È pertanto semplice usare una tabella di ramificazione, o qualunque altra tecnica di conversione, per ottenere il codice corrispondente alla chiave. Inoltre, se più di uno zero è presente sia nel primo «nibble» (gruppo di 4 bit) che nel secondo, esso rivela una *chiusura di più tasti*, cioé un problema di *rollover*. Esso è trattato normalmente attraverso la tabella di salto. Un tale codice, avendo però zeri illegali, produce una ramificazione su un ingresso di tabella non valido. Esso può essere rivelato oppure pu ichiedere che l'intero processo sia rieseguito,

ignorando pertanto l'ingresso finché sia premuto un solo tasto.

Il vantaggio di tale tecnica è di richiedere un programma software molto semplice, e di eliminare la circuiteria necessaria per scandire le righe. Lo svantaggio è di dedicare una porta del PIO alla gestione della tastiera. Tuttavia, tenendo conto del costo molto basso dei PIO, la soluzione proposta risulta economica.

Tastiera codificata

Non è cosa piacevole scrivere il software per la codifica della tastiera. Diversi tipi di circuiti di interfaccia LSI sono usati per codificare le tastiere. Normalmente il circuito scandisce la matrice, scopre la coincidenza, provvede al rollover e allo smorzamento, controlla il trasferimento dei dati da usare nel sistema. Alcune unità forniscono anche tabella interna ROM di controllo per generare il segnale codificato relativo al tasto premuto, diverso secondo il tipo di codice, ASCII, EBCDIC, etc.

Con questo singolo circuito integrato e con il sistema microcalcolatore è realizzata un'interfaccia completa di *generazione e di visualizzazione.* È possibile infatti osservare, in figura 4-13 che l'8279 costituisce l'intera sezione di generazione e di visualizzazione di un terminale per punto di vendita che usa il sistema microcalcolatore 8048 a circuito integrato singolo.

Codificatori di tastiera

Il ruolo fondamentale del codificatore di tastiera è di identificare il tasto che è stato premuto e di produrre il codice a 8 bit del tasto che gli corrisponde. Un buon circuito integrato di tastiera deve anche risolvere i problemi che abbiamo già indicato. Esso deve cioé eliminare gli effetti del *rimbalzo* e fornire la protezione al *rollover.* Sono disponibili tre tipi fondamentali di codificatori: codificatori *statici*, a *scansione* e *convertitori.*

Un *codificatore statico* genera semplicemente il codice corrispondente al tasto. Per semplificare il problema della protezione del tasto è presa in considerazione una tastiera lineare. Una tastiera lineare è, per esempio, una tastiera a 64 tasti che ha un collegamento rigido per ogni tasto premuto. La rivelazione è pertanto semplice. L'impulso apparirà nella connessione relativa al tasto premuto. Questo impulso è semplicemente trasformato nel codice a 8 bit desiderato. Ciò significa, tuttavia, 64 vie separate di ingresso per produrre un codice di 64 x 8 bit. Per ridurre il costo della connessione e il costo dei codificatori molte tastiere sono organizzate a matrice, per esempio 8 x 8. In una tastiera 8 x 8 sono usate soltanto 16 connessioni. Il prezzo da pagare è che il processo necessario per identificare il tasto diventa piú complesso. È cioé necessario un *codificatore a scansione*, o l'uso di una *routine di scansione.* Costose tastiere ASCII (solo tastiere) possono permettersi il lusso di una struttura lineare in base al costo di ciascun tasto. Non è cioé necessario identificare il tasto. La maggior parte delle tastiere hanno però la struttura a matrice.

Circuito integrato di scansione

Il *circuito integrato di scansione* risolve il problema dell'identificazione del tasto, quando si usa un'organizzazione a matrice della tastiera. È scandita una riga di tasti alla volta mediante un contatore. Finché non è premuto nessun tasto è realizzata una scansione circolare. Appena uno di essi è premuto, è generato un impulso indicante la chiusura del contatto di un tasto ed è bloccata la scansione. La lettura dello stato del contatore rivela la riga e la colonna del tasto che è stato premuto. Un tale meccanismo di scansione lineare non garantisce la protezione del rollover a due tasti. Con questo sistema la scansione si ferma al primo tasto che trova premuto. Quando sono premuti due tasti quasi contemporaneamente può essere rivelata la battuta del secondo tasto prima di quella del primo. Un meccanismo migliore aspetterà la scansione dell'intera tastiera e genererà un codice valido solo se è rivelata la battuta di un solo tasto. Quando rivelerà la battuta di più di un tasto, continuerà a scandire finché un solo tasto è premuto. Questo metodo ha il vantaggio intrinseco di fornire insensibilità automatica ai rimbalzi del tasto.

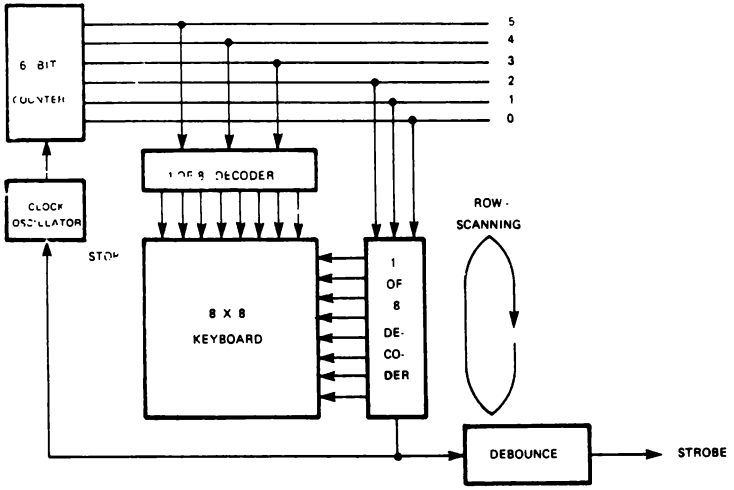


Fig. 4-8: Scansione di tastiera

Tutti i problemi trattati sono infatti semplificati. Per la rivelazione della battuta del tasto è necessario fornire la tensione alle colonne e, se esse fossero tutte attivate nello stesso tempo, sarebbe impossibile determinare quale colonna è stata premuta. In realtà la tensione è fornita ad una colonna alla volta e, quando è rivelata una battuta di tasto, è nota la colonna e le righe sono scandite per rivelare un'altra battuta.

Il funzionamento della scansione è normalmente il seguente: è usato un solo contatore a 6 bit. I suoi tre bit più significativi sono decodificati da un decodificatore 1 ad 8 e sono usati per decodificare ciascuna delle otto colonne in sequenza. I tre bit meno significativi del contatore, che cambiano più frequentemente il loro valore, sono anche loro decodificati da un decodificatore per scandire le 8 righe in sequenza. Per otto righe scandite è scandita una colonna.

Quando si verifica la chiusura di un tasto la rivelazione si verifica quando è selezionata la riga. Ciò comporta l'arresto del contatore a 6 bit. Sono allora letti i contenuti del contatore per identificare la riga e la colonna del tasto premuto.

Buoni codificatori di tastiera dispongono di una ROM che fornisce automaticamente il codice di uscita del tasto premuto. Esse dispongono anche di ingressi separati di scorrimento (shift) e controllo che eliminano codici di uscita falsi quando sono premuti tasti in modo non decodificabile.

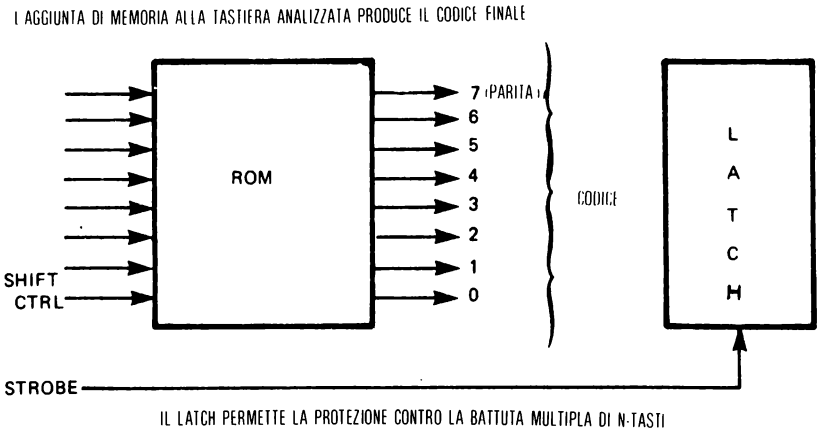


Fig. 4-9: ROM e controllo trasferimento («latch»)

In figura 4-10 è indicato, a titolo di esempio, il codificatore di tastiera NECuPD 364D–02.

Esso fornisce il controllo su n tasti, rollover di n tasti e insensibilità alle oscillazioni, controllo della frequenza dell'oscillatore, e la selezione di 4 modi di funzionamento: scorrimento, controllo, e scorrimento più controllo.

Contiene una ROM per bit di tipo 3600 con 10 bit di uscita per ciascuno dei 90 tasti e funzionante nei 4 diversi modi indicati. I 90 tasti della tastiera possono essere organizzati in matrice 9x10. È equipaggiata con un contatore ad anello a 10 stadi per le colonne, e di un altro a 9 stadi per le righe. Inoltre la sua memoria contiene in uscita una memoria temporanea per i dati. Essa garantisce che non ci siano in u-

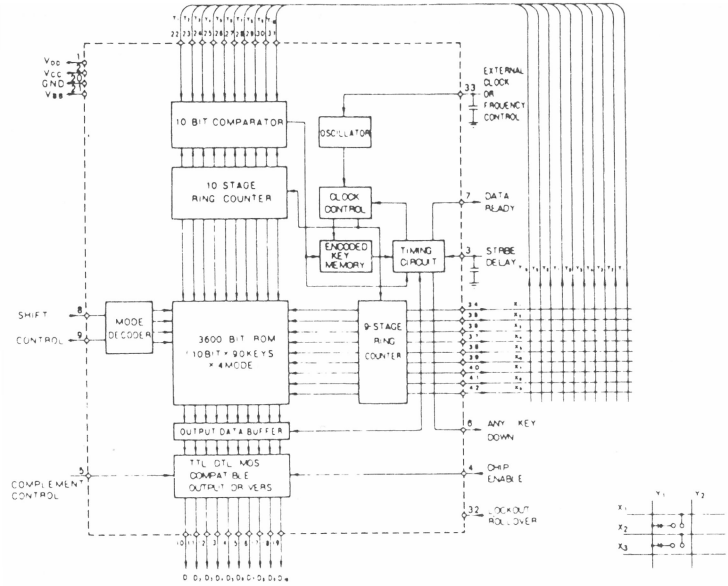


Fig. 4-10: Codificatore di tastiera NEC

scita codici casuali, mentre è eseguita la scansione e non sono premuti tasti. Codificatori simili sono disponibili da diversi costruttori, come la General Instrument e altri. La ROM in unico circuito integrato può essere programmata mediante mascheratura per ottenere il codice di tipo desiderato, come l'ASCII o lo EBCDIC.

Questo componente può essere usato in un sistema a microprocessore come porta di ingresso attraverso il bus. La via «dato pronto» può essere usata come semaforo per il processore quando una battuta di tasto è pronta per essere letta.

Fig. 4-11: Tastiera ASCII

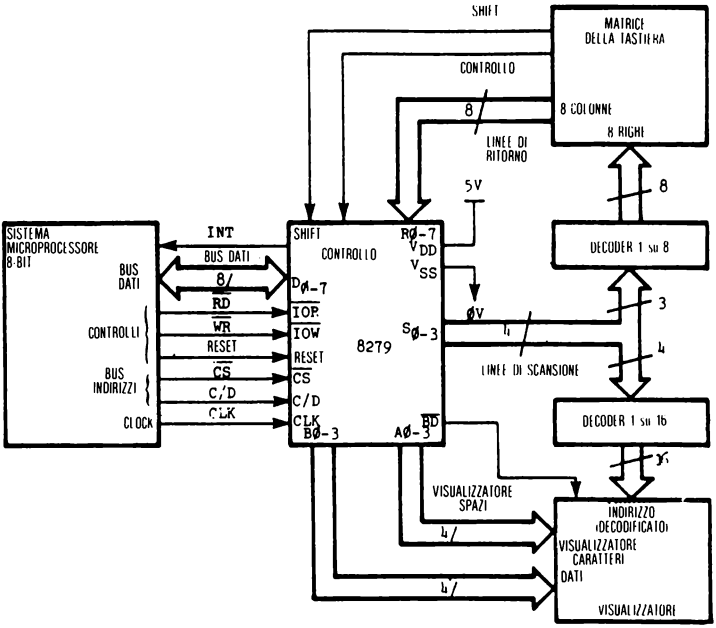


Fig. 4-12: Controllore di schermo di tastiera 8279

Il circuito LSI indicato in figura 4-12 dispone di una matrice 8 x 8 per la tastiera, con terminazioni di controllo e scorrimento. In tal modo è possibile generare fino a 256 codici diversi. Per esempio, premendo «controllo» e «scorrimento» e la lettera «p» risulta uno dei codici disponibili.

Oltre a codificare la tastiera, il componente scandisce e comanda la visualizzazione di uno schermo, per presentare i dati memorizzati in una banca RAM nell'8279. Componenti simili sono disponibili dalla Rockwell e dalla GI.

Tastiere ASCII

Le tastiere possono essere acquistate con lo schema standard di teletype o di macchina da scrivere per generare il codice ASCII a sette bit. Queste tastiere contengono i tasti oltre al circuito integrato LSI di controllo della tastiera. L'uscita è normalmente a 7 bit in parallelo con un impulso di campionamento (strobe). Per interfacciare verso un ingresso seriale standard, é aggiunta una UART e un circuito di temporizzazione (clock). Un progetto completo é indicato in figura 4-15.

Lo UART preleva i dati a 7 bit e li trasmette in un formato seriale a 10 o 11 bit quando é presente l'impulso di battuta. La tastiera é controllata nella sua uscita mentre trasmette. Il segnale di temporizzazione (clock) seriale ha una frequenza 16

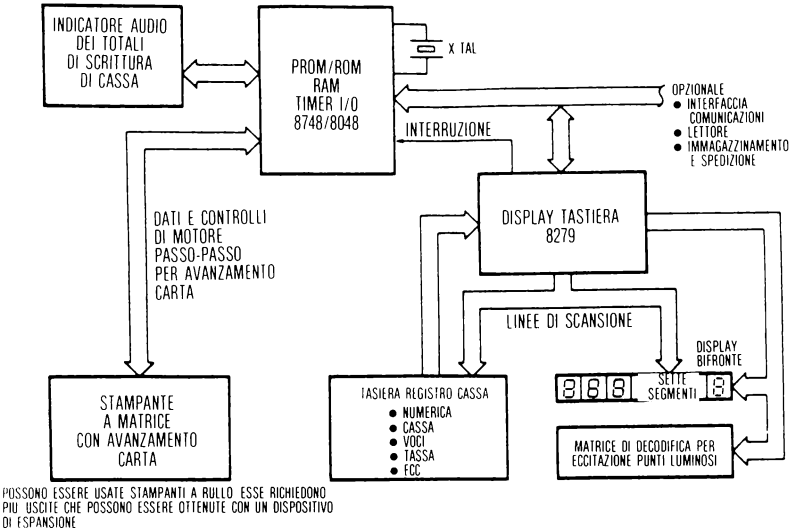


Fig. 4-13: Terminale per punto di vendita 8048 con l'8279

NUMERI DEI BIT															
								0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
b_7	b_6	b_5	b_4	b_3	b_2	b_1	HEX 1 / HEX Ø	0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
			0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
			1	0	0	0	8	BS	CAN	(	8	H	X	h	x
			1	0	0	1	9	HT	EM	)	9	I	Y	i	y
			1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	11	VT	ESC	+	;	K	[	k	{
			1	1	0	0	12	FF	FS	,	<	L	\	l	¦
			1	1	0	1	13	CR	GS	-	=	M	]	m	}
			1	1	1	0	14	SO	RS	.	>	N	^	n	~
			1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Fig. 4-14: Tabella ASCII

volte superiore a quella della velocità di trasmissione dei bit. Per 110 baud, l'oscillatore é accordato a 1760 Hz. Per 300 baud esso é accordato a 4800 Hz.

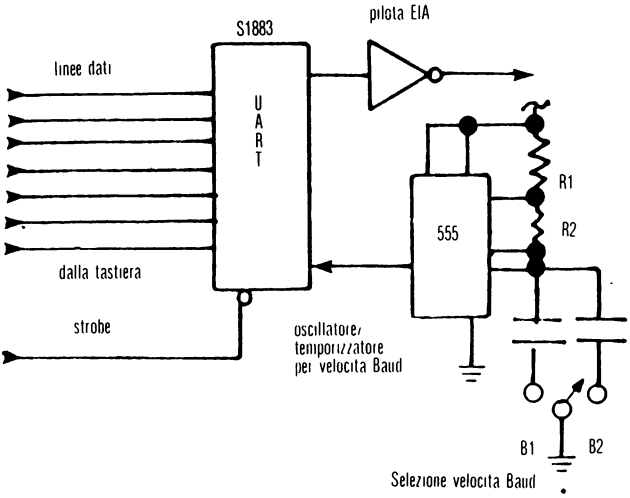


Fig. 4-15: Interfaccia di tastiera seriale ASCII

SCHERMI LED

I diodi emettitori di luce (LED) sono comunemente usati per indicare lo stato o altre informazioni all'utente. Gli schermi LED possono avere diverse forme. Tre di esse sono: LED singolo, LED a sette segmenti, e schermo LED a matrice di punti.

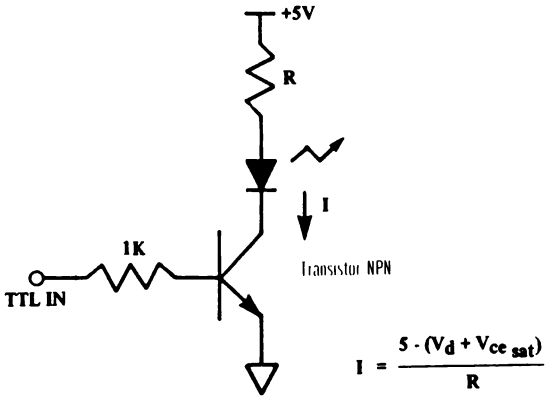


Fig. 4-16: Interfaccia di LED singolo

Il LED *singolo* é un diodo con un salto di tensione da 1,2 a 2,4 volt, secondo il tipo. Esso é un componente che emette una luce visibile o infrarossa su una stretta banda di lunghezze d'onda. I LED più usati sono quelli rossi. Altri usati, anche se più costosi e talvolta non di pari efficienza, sono il verde, l'arancio, il giallo, e l'infrarosso.

In figura 4-16 é indicata una interfaccia LED verso una porta di uscita a singolo bit.

La corrente, I, che passa attraverso il LED determina la sua intensità luminosa. La formula indicata può essere semplificata in: $I = 3,5/R$ per una alimentazione di 5 volt. Correnti tipiche sono da due a venti milliampere. Quando l'ingresso é minore di 0,6 volt, il transistore é «off» e non scorre corrente. Quando l'ingresso é maggiore di 0,6 volt, il transistore commuta su «on» e permette che la corrente scorra, rendendo il LED luminoso.

LED a sette segmenti

Uno schermo LED a sette segmenti consiste di un gruppo di 7 LED elementari organizzati come in figura 4-17.

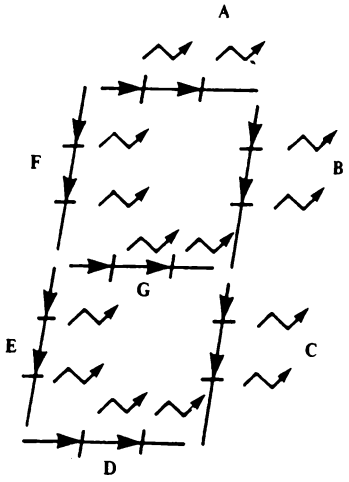


Fig. 4-17: LED a 7 segmenti

Con questi segmenti é possibile visualizzare i numeri da 0 a 9 ed alcune lettere dell'alfabeto. In tal modo si dispone di una *uscita visualizzata* dei sette segnali di pilotaggio.

Un componente di interfaccia comune é il circuito di *pilotaggio e decodifica da BCD a sette segmenti.* Esso converte un BCD a 4 bit direttamente nei numeri ap-

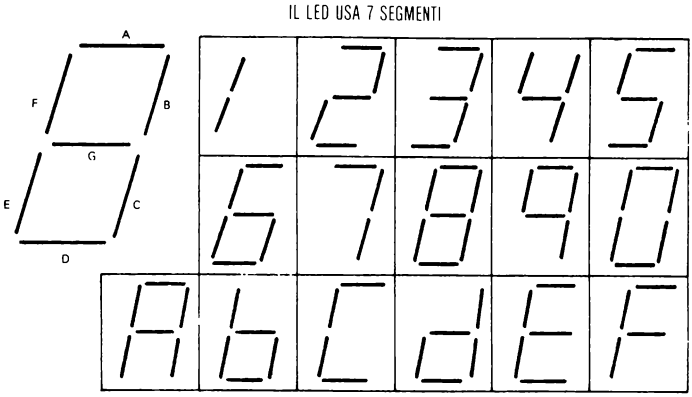


Fig. 4-18: Caratteri a 7 segmenti

propriati e anche pilota i LED con transistori di pilotaggio interni. Un esempio é il 7447 indicato in figura 4-19. Una porta di uscita può pilotare un 7447 con 4 bit del dato BCD per eccitare i segmenti opportuni. In figura 4-20 é indicata la tabella della verità.

Per ridurre il costo di un decodificatore per ciascuno schermo LED, il segnale di visualizzazione può essere multiplato. Ciascun «digit» é abilitato per un breve tempo prima che un altro sia selezionato e abilitato. In tal modo un decodificatore può servire per un certo numero di schermi. Ci sono diversi modi per multiplare e due di essi sono di seguito indicati.

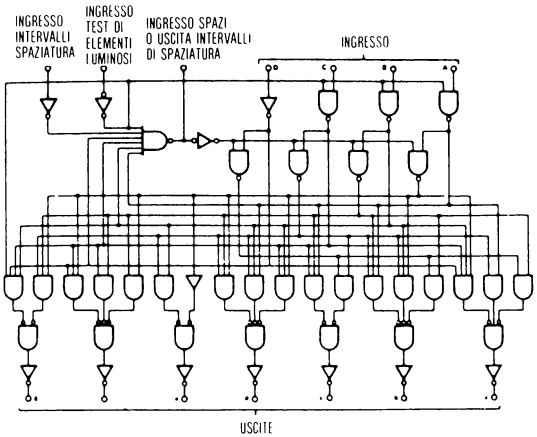


Fig. 4-19: Pilotaggio-decodifica verso il sette segmenti 7447

TABELLA DELLA VERITA'

	INGRESSI							USCITE							
DECIMALE O FUNZIONE	LT	RBI	D	C	B	A	BI/RBO	a	b	c	d	e	f	g	NOTA
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1
1	1	x	0	0	0	1	1	1	0	0	1	1	1	1	1
2	1	x	0	0	1	0	1	0	0	1	0	0	1	0	
3	1	x	0	0	1	1	1	0	0	0	0	1	1	0	
4	1	x	0	1	0	0	1	1	0	0	1	1	0	0	
5	1	x	0	1	0	1	1	0	1	0	0	1	0	0	
6	1	x	0	1	1	0	1	1	1	0	0	0	0	0	
7	1	x	0	1	1	1	1	0	0	0	1	1	1	1	
8	1	x	1	0	0	0	1	0	0	0	0	0	0	0	
9	1	x	1	0	0	1	1	0	0	0	1	1	0	0	
10	1	x	1	0	1	0	1	1	1	1	0	0	1	0	
11	1	x	1	0	1	1	1	1	1	0	0	1	1	0	
12	1	x	1	1	0	0	1	1	0	1	1	1	0	0	
13	1	x	1	1	0	1	1	0	1	1	0	1	0	0	
14	1	x	1	1	1	0	1	1	1	1	0	0	0	0	
15	1	x	1	1	1	1	1	1	1	1	1	1	1	1	
BI	x	x	x	x	x	x	0	1	1	1	1	1	1	1	2
RBI	1	0	0	0	0	0	0	1	1	1	1	1	1	1	3
LT	0	x	x	x	x	x	1	0	0	0	0	0	0	0	4

Fig. 4-20: Tabella di verità del 7447

La figura 4-21 indica il primo schermo, che scandisce sia «digit» che dati. È da osservare che sono usati *circuiti di pilotaggio esterni.* Ciò avviene perché, quando multiplato, uno schermo deve essere N volte più brillante rispetto a quando opera da solo, poiché é abilitato per un ennesimo del tempo. Pertanto é necessaria una corrente N volte superiore. Gran parte dei circuiti integrati non possono erogare tale corrente, e pertanto devono essere usati transistori discreti esterni.

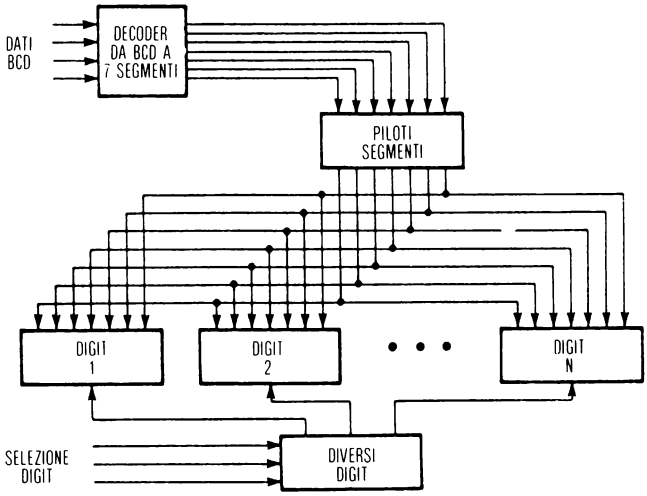


Fig. 4-21: Multiplazione di LED

Il secondo schema, in figura 4-22, usa un *contatore* per fare avanzare il contatore di «digit». Il conteggio é presentato in uscita verso il processore e usato per indirizzare il dato appropriato per il digit. Il dato é posto su una porta di uscita che pi-

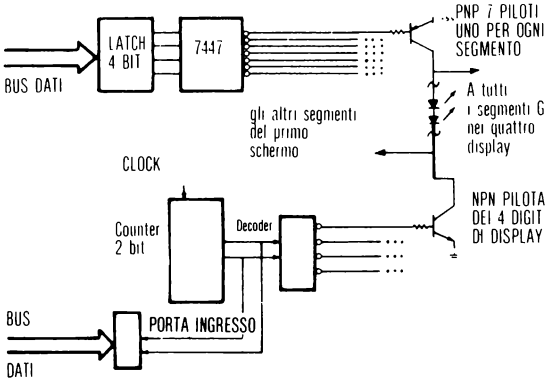


Fig. 4-22: Pilotaggi per la scansione

lota il decodificatore 7447. È ancora da notare che é necessario un circuito erogatore di corrente per aumentare la luminosità.

LED a matrice

Il LED a matrice consiste di 5 righe e di sette colonne nelle cui intersezioni sono posti i LED. Questi 35 LED possono rappresentare lettere maiuscole e minuscole, oltre che numeri. Una organizzazione tipica é rappresentata in figura 4-23

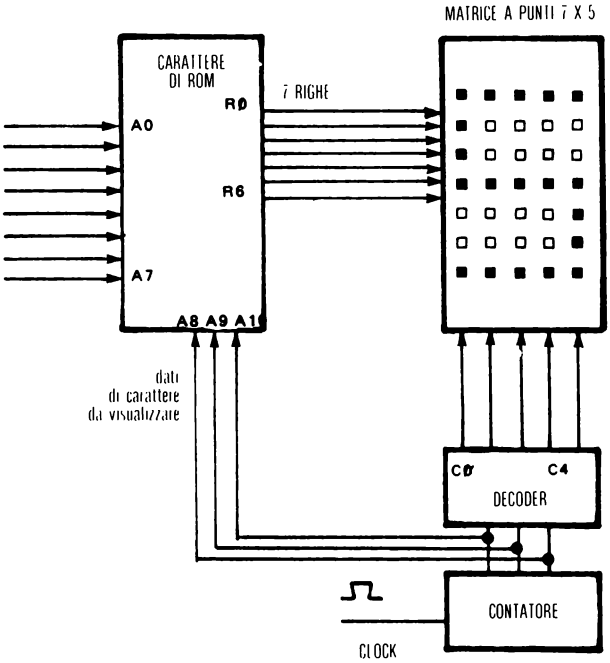


Fig. 4-23: LED a matrice di punti 7x5

La prima porta di uscita seleziona i dati di colonna, mentre la seconda seleziona le righe, attraverso il decodificatore. Con questa tecnica il programma procede passo passo lungo le cinque righe, visualizzando ogni qual volta è stato programmato un carattere nella ROM 2048 x 8.

Altra tecnica è di avere una scansione lungo le righe mediante hardware esterno e visualizzazione del dato opportuno. Tale metodo è illustrato in figura 4-24.

Il contatore conterà da zero a quattro. La ROM di carattere si suppone indirizzata al carattere «S». La colonna 0 indirizza i dati di riga da visualizzare. Essi sono da R6 a RQ: 1001111_2. Il clock fa avanzare il contatore alla colonna 1. I dati di riga

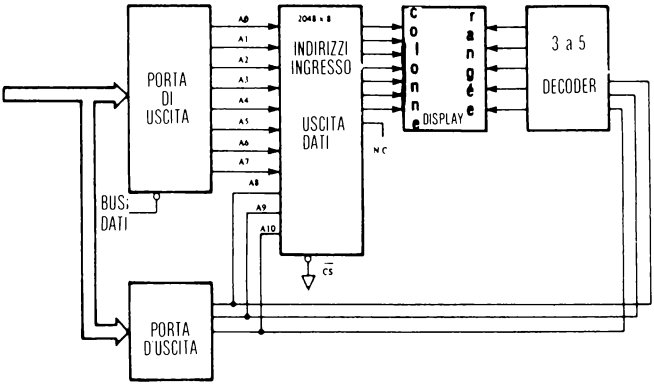


Fig. 4-24: Matrice LED a contatore multiplato

sono ora 1001001_2. Ciò continua fino alla colonna 4 e poi si ripete. In tal modo: possono essere generate tutte le lettere dell'alfabeto. Una ROM tipica per caratteri è rappresentata in figura 4-25. L'esempio rappresentato è uno schermo 7x9 per una risoluzione più spinta. Inoltre la ROM per carattere indicata può essere usata con codice ASCII, Baudot o EBCDIC, in relazione alla tabella ordinata.

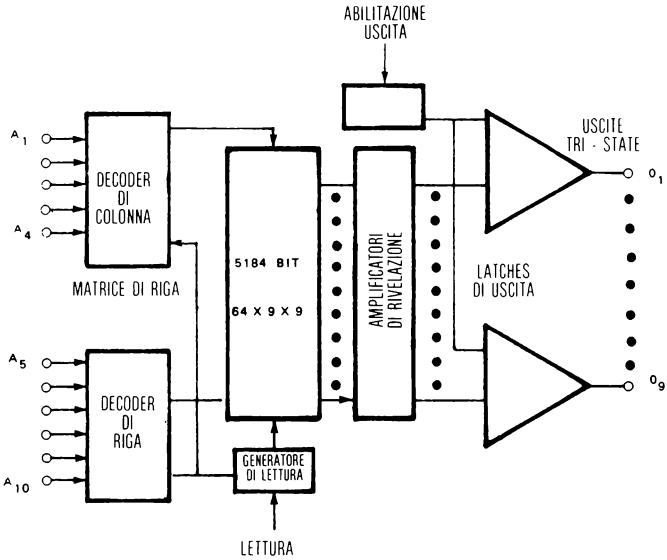


Fig. 4-25: ROM a matrice di punti

Conclusioni sugli schermi

Esistono molti altri schermi. In generale, gli schermi di tipo LED sono affidabili, facili da interfacciare, e rispecchiano le tecniche usate in gran parte di tutte le altre tecniche di interfaccia per schermi. Le tecniche di interfacciamento dei CRT saranno trattate anche in questo capitolo, e il metodo di matrice a punti sarà indicato in quella sezione.

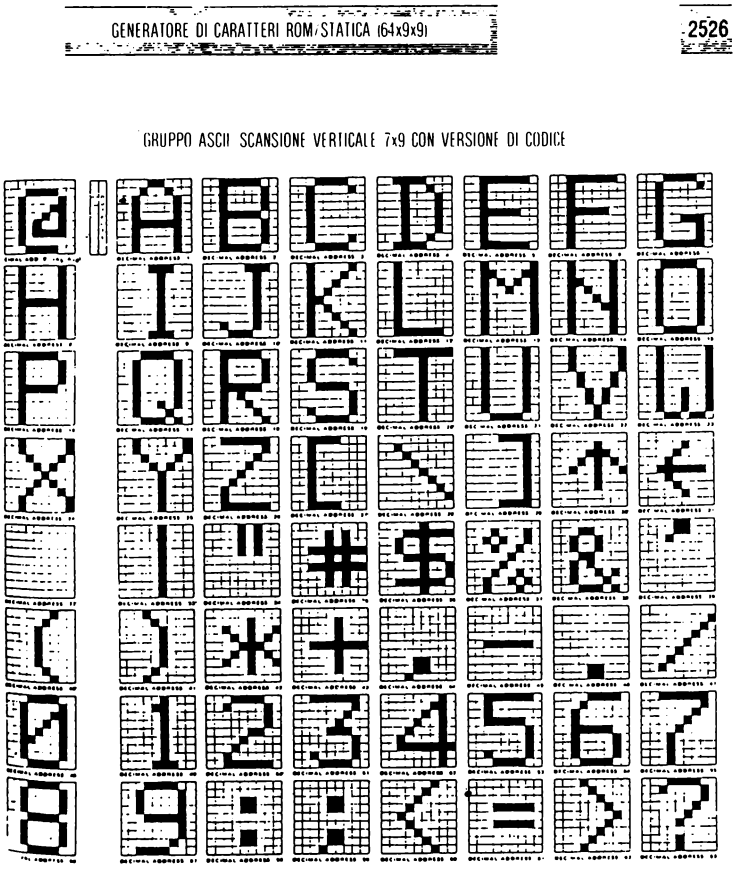


Fig. 4-26: Caratteri a matrice di punti

TELETYPE

Una teletype è un meccanismo di ingresso/uscita seriale che opera normalmente a 110,150 o 300 baud, secondo il modello e il costruttore. Sono qui presentati tre metodi di interfacciamento: uno per una UART, per una Teletype® modello 33,

uno per un'ACIA Motorola, per una Teletype® modello 33 con opto-isolatori, e uno per l'interfaccia RS232EIA. Una Teletype modello 33 opera a 10 caratteri al secondo. Ciascun carattere è codificato con 11 bit: un bit di start, 8 bit di dati, e 2 bit di stop. La velocità di trasferimento è pertanto 110 Baud. L'unico problema di interfacciamento significativo è assemblare il byte di dati a 8 bit in parallelo dagli 11 bit. La trasmissione è asincrona. Ma interfaccia universale per una TTY è lo UART, che è stato descritto nella precedente sezione. Esso realizza automaticamente tutte le funzioni richieste, e può operare in entrambe le direzioni.

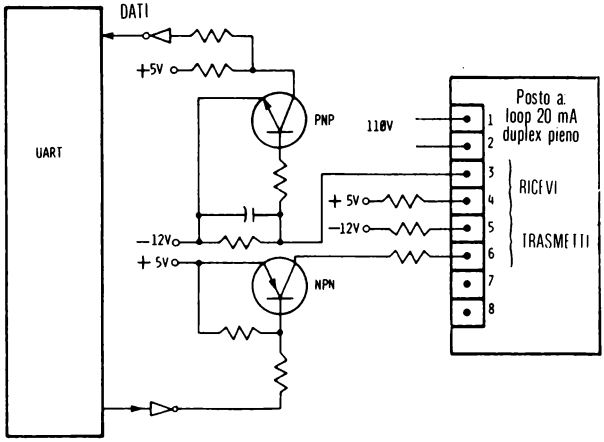


Fig. 4-27: Interfaccia tra UART e TTY

In figura 4-27 lo UART è usato per la conversione dei dati da serie a parallelo, e viceversa. La figura 4-28 indica il formato seriale, mentre la 4-29 indica la sequen-

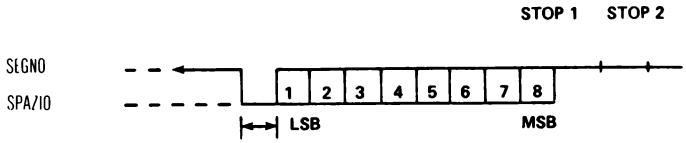


Fig. 4-28: Formato seriale di dati

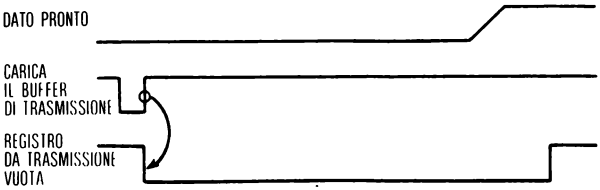


Fig. 4-29: Temporizzazione UART

za di campionamento. Lo schema della interfaccia indica come i segnali TTL sono convertiti in segnali a loop di corrente di 20 mA richiesti dalla TTY.

In figura 4-30 sono usati *opto-isolatori* per isolare elettricamente la teletype dal sistema microcalcolatore. *Ciò richiede che i livelli di + e − 12 volt siano anche isolati dal microcalcolatore.* L'ACIA realizza la conversione e interfaccia direttamente con il bus 6800.

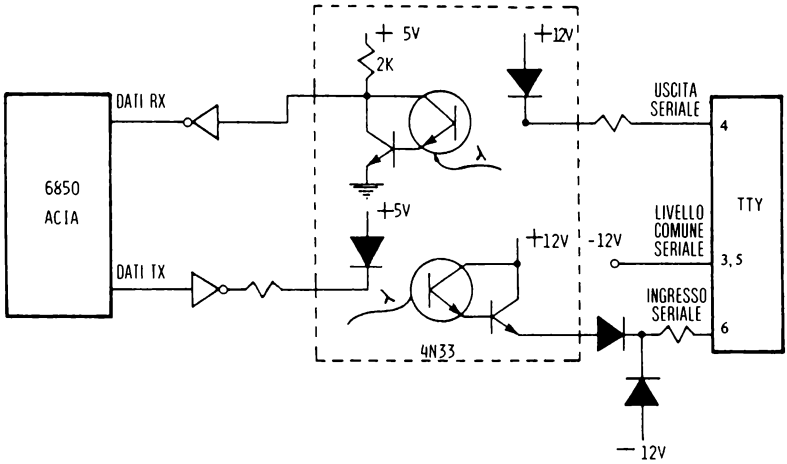


Fig. 4-30: Interfaccia verso la TTY opto-isolata

Alcune teletype sono provviste dello EIA-RS232C in una configurazione seriale. In teletype RS232C, sono usati impulsi di + o − 12 volt piuttosto che la presenza o assenza di correnti di 20 milliampere. La figura 4-31 indica l'insieme di componenti necessari per la conversione da EIA a TTL e da TTL a EIA. Essi sono l'MC1489 e l'MC1488. Essi contengono 4 traslatori in ciascun contenitore, potendo così interfacciare più linee.

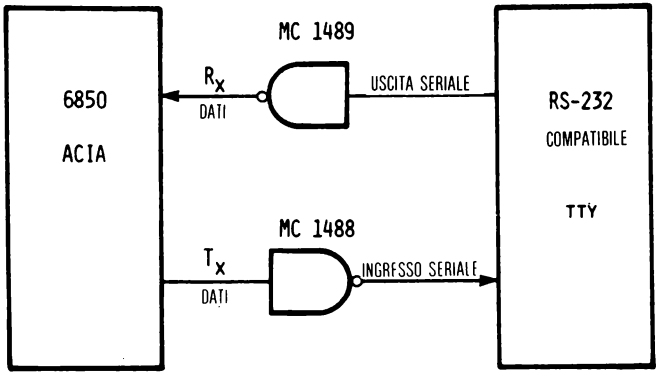


Fig. 4-31: Integrati EIA interfaccianti con MC1488 e MC1489

```
NEXT 1    LDA A STACON      LOAD STATUS
          ASR A             SHIFT RDRF BIT TO C-BIT POSITION
          BCS FRAM          CHECK RDRF BIT
          ASR A
          ASR A             SHIFT DCD BIT TO C-BIT POSITION
          BCC NEXT 1        CHECK DCD BIT
          BR ERROR 2        CARRIER LOSS - BRANCH TO ERROR ROUTINE
FRAM      ASR A
          ASR A             SHIFT FE BIT TO C-BIT POSITION
          BCC OVRN          CHECK FE BIT
          BR ERROR 3        FRAMING ERROR - BRANCH TO ERROR ROUTINE
OVRN      ASR A             SHIFT OVRN BIT TO C-BIT POSITION
          BCC PAR           CHECK OVRN BIT
          BR ERROR 4        OVERRUN ERROR - BRANCH TO ERROR ROUTINE
PAR       ASR A             SHIFT PE BIT TO C-BIT POSITION
          BCC R DATA        CHECK PE BIT
          BR ERROR 5        PARITY ERROR - BRANCH TO ERROR ROUTINE
R DATA    LDA B TXRX        LOAD B REGISTER WITH DATA
          RTS               RETURN FROM SUBROUTINE
```

Fig. 4-32: Subroutine di ricezione del 6800

Dal punto di vista meccanico la teletype sembra complessa, ma in realtà è semplice. Per agevolare la comprensione del formato dei dati in serie, sarà indicato cosa avviene internamente.

Quando si presenta il bit di start, succedono due cose: l'innesto aggancia tutti gli accoppiamenti meccanici così da iniziare un ciclo di stampa, e prepara il magnete selettore di decodifica per il processo di decodifica. 9,09 millisecondi dopo si presentano gli 8 bit successivi. Ciascuno di essi libera il magnete del selettore, che blocca la rotazione di ruote a 8 denti, per otto volte in sequenza. Nel frattempo, sono sollevate o abbassate le barre di stampa che selezionano il carattere nella testa di stampa. Ciò in relazione alla combinazione dei denti della ruota. La testa di stampa seleziona il carattere opportuno e il martelletto di stampa colpisce la carta e il rullo. I bit di stop sono necessari per aver tempo sufficiente per completare la stampa del carattere in corso, prima che se ne presenti un altro. Se è attivato anche il perforatore, la selezione delle barre di stampa produrrà anche fori sul nastro perforato, durante la stampa del carattere.

Quando un tasto è premuto, l'opportuna trama del bit è posta in otto contatti nel *distributore.* Il distributore è simile al suo omonimo nelle automobili. La figura 4-33 indica la semplicità dello schema. Il motore è impegnato a far ruotare il commutatore, aprendo e chiudendo i contatti e generando così la trama a 11 bit per quel tasto.

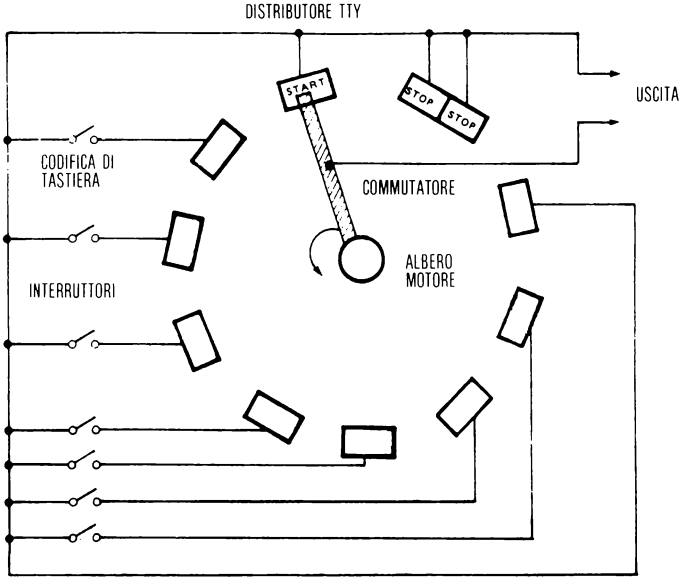


Fig. 4-33: Distributore nella Teletype

È da notare che il motore sincrono è la sorgente di temporizzazione della macchina, e che pertanto è necessaria una frequenza di linea precisa. La macchina perderebbe altrimenti sincronismo, sia per vecchiaia, assenza di olio, o altri problemi meccanici.

Una subroutine di uscita di teletype

Si assume qui che la teletype sia connessa al bit 0 della porta 2. Questo semplice programma farà scorrere in uscita gli 11 bit necessari per presentare il carattere in formato teletype. Il diagramma di flusso è indicato in figura 4-34, mentre le connessioni sono indicate in figura 4-36. Il programma è di seguito spiegato. Il registro B è usato come contatore ed è inizialmente posto ad 11. I contenuti del registro B sono decrementati ogni volta che il bit è fatto scorrere fuori, cioè trasmesso alla porta 2. Questo è importante per ricordare che soltanto il bit 0 dell'accumulatore ha importanza in questo esempio. Tutti gli altri bit sono ignorati. Esso è il bit più a destra, o bit meno significativo (LSB) dell'accumulatore. Inizialmente l'accumulatore contiene gli 8 bit da trasmettere. La trasmissione di entrambi i bit di start e stop è realizzata usando l'istruzione di rotazione dell'8080. Il bit di riporto (carry), che è il nono bit dell'accumulatore, nella operazione di scorrimento è posto a zero. Esso è pertanto ruotato nella posizione 0 dell'accumulatore e sarà il bit di start. Il nócciolo dell'operazione è l'uso della istruzione *rotazione*. Se i contenuti dell'accumulatore si

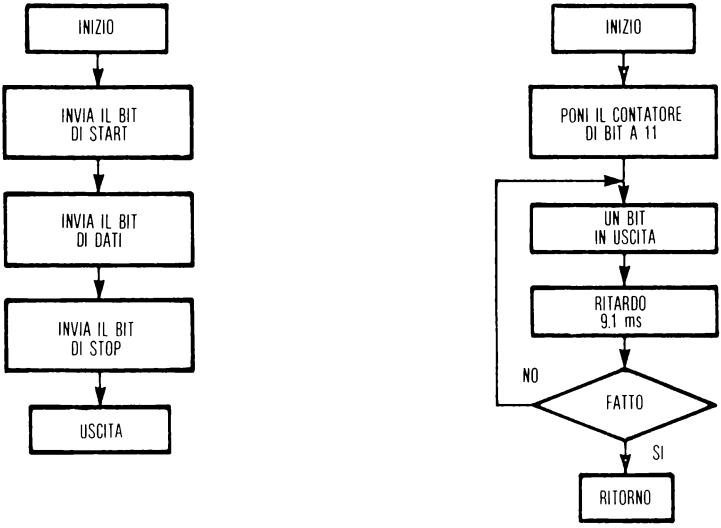


Fig. 4-34: Software di trasmissione

```
TELETYPE OUTPUT SUBROUTINE          (ASSUME TTY CONNECTED TO PORT 2 BIT
                                                                    0)
        ;
        ;  THIS SUBROUTINE ENTERED WITH CHARACTER TO BE OUTPUT IN THE
        ;                                                   C REGISTER
        TYOUT:   MVI    B,11     ;  SET COUNTER FOR 11 BITS
                 MOV    A,C      ;  CHARACTER TO ACCUMULATOR
                 ORA    A        ;  CLEAR CARRY-FOR START BIT
                 RAL             ;  MOVE CARRY TO A(0)
        MORE:    OUT    2        ;  SEND TO TTY
                 CALL   DELAY    ;  KILL TIME
                 RAR             ;  POSITION NEXT BIT
                 STC             ;  SET CARRY-FOR STOP BITS
                 DCR    B        ;  DECREMENT BIT COUNTER
                 JNZ    MORE     ;  DONE?
                 RET             ;  YES
        ;
        ;  9 MSEC DELAY (ASSUME NO WAIT STATES)
        ;
        DELAY:   MVI    D,6
        DL0:     MVI    E,2000
        DL1:     DCR    E        ;  1.5 MSEC
                 JNZ    DL1      ;  INNER LOOP
                 DCR    D
                 JNZ    DL0
                 RET
```

Fig. 4-35: Programma di uscita verso la TTY per l'8080

facessero *scorrere* semplicemente a sinistra, si perderebbe il bit alla estrema sinistra. In tal modo il bit all'estrema sinistra è preservato nel carry, mentre un bit 0 risulta scritto nella posizione 0. È da notare nel programma che l'ultima operazione eseguita sull'accumulatore è una rotazione a destra. Essa ripristina il precedente bit 7, che è stato preservato nel bit di carry, nella sua corretta posizione. Una volta che questo è stato fatto, successive rotazioni ruotano a sinistra dell'accumulatore uni successivi creati nel bit carry. Ciò garantisce che sia trasmesso il bit di stop alla fine. La sequenza del programma è lineare:

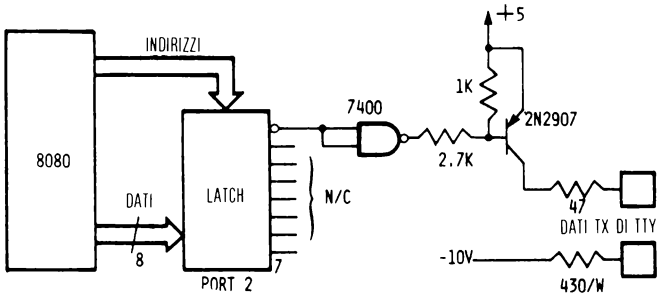


Fig. 4-36: Interfaccia hardware verso TTY

Il registro contatore B è posto al valore 11, mentre il carattere che è stato preservato nel registro C è caricato nell'accumulatore A. L'accumulatore è messo in or con sé stesso (terza istruzione). Questo non cambia i suoi contenuti, ma garantisce che il carry sia posto a 0. Esso sarà il bit di start. È realizzata una rotazione a destra: RAR. Questo sposta il carry nella posizione 0 dell'accumulatore. È infine eseguita l'uscita: OUT 2. Il bit è spedito alla teletype. Ogni volta che un bit è spedito alla teletype deve essere garantito un ritardo di 9 ms. La routine di ritardo è realizzata mediante una subroutine, indicata nella parte finale del programma. Successivamente è eseguita una RAR per far scorrere nella posizione di bit 0 il bit successivo corretto. Il carry è posto ad 1 prima di ulteriori rotazioni per garantire che l'eventuale bit di start sia trasmesso correttamente. Il contatore di bit (registro B) è infine decrementato e controllato. Se il contatore raggiunge il valore 0 il programma termina, altrimenti il programma esegue un «loop» tornando indietro all'indirizzo MORE, dove ha inizio l'uscita successiva.

Esempio di software per l'ACIA

La subroutine che stiamo per descrivere manda un carattere alla teletype. Se essa non è pronta a trasmettere, la subroutine *attende* finchè è pronta. Essa controlla anche l'ingresso «abilitazione a trasmettere» (clear to send-CTS) nell'ACIA. È da usare con il sistema di interfaccia EIA-RS232C.

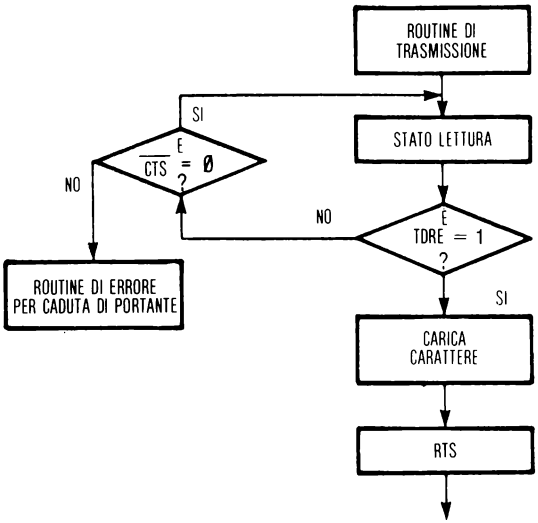


Fig. 4-37: Diagramma di flusso dell'ACIA

La prima istruzione carica lo stato dell'ACIA nell'accumulatore A. Il semaforo «pronto a trasmettere» è nel bit di posizione 1, e può pertanto essere fatto scorrere due volte verso destra, per essere controllato. Se si è pronti a trasmettere il programma va direttamente a DATI, e il contenuto dell'accumulatore B è spedito all'ACIA.

Se l'ACIA non è pronta a trasmettere, è controllato il bit CTS; se esso indica abilitazione, si è verificata una perdita di portante e il programma si ramifica su una routine di errore. Se l'ACIA è pronta a trasmettere, il semaforo pronto a trasmettere è controllato finchè indica abilitazione. Questa è una tecnica polling. Può anche essere usata una interruzione.

```
NEXT     LDA A STACON        LOAD STATUS
         ASR A
         ASR A               SHIFT TDRE BIT TO C-BIT POSITION
         BCC TX DATA         CHECK TDRE BIT
         ASR A
         ASR A               SHIFT CTS BIT TO C-BIT POSITION
         BCC NEXT            CHECK CTS
         BR ERROR 1          CARRIER LOSS - BRANCH TO ERROR ROUTINE

TX DATA  STA B TXRX          STORE CHARACTER IN ACIA
         RTS                 RETURN FROM SUBROUTINE
```

Fig. 4-38: Subroutine di trasmissione dell'ACIA

LETTORE DI NASTRO PERFORATO

Le telescriventi sono spesso lente per la lettura del nastro perforato. Una periferica utile è il lettore di nastro perforato ad alta velocità. Tale componente rivela otticamente la trama del codice perforato nel nastro e avanza rapidamente. Lo schema di un lettore tipico è rappresentato in figura 4-39.

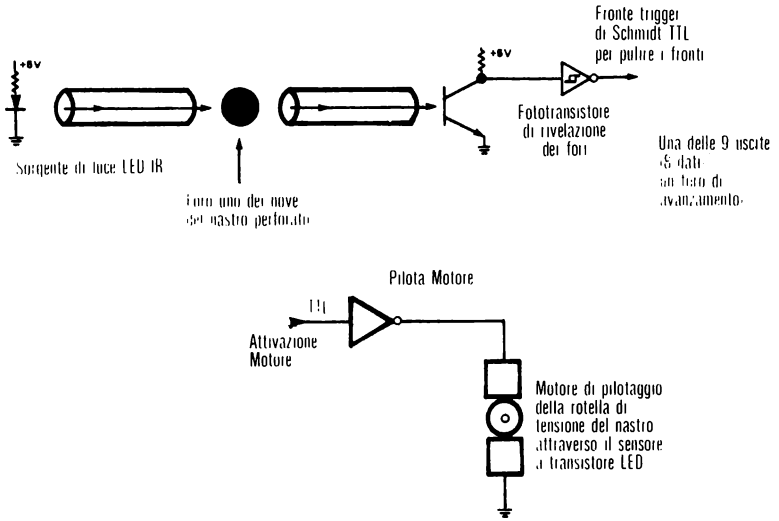


Fig. 4-39: Lettore di nastro perforato

Un microcalcolatore attiva il motore, cerca il foro di avanzametro (che è più piccolo dei dati, indicante il centro della trama dei bit), legge la sequenza di trama, e memorizza il dato prima di dover considerare il foro di avanzamento successivo. Quando è rivelato il carattere fine del nastro, il motore che controlla la lettura è arrestato.

In figura 4-40 è indicata la trama dei bit di un nastro a 8 livelli. Un problema tipico è rappresentato dai bordi del foro non perfettamente centrati o da sporcizia

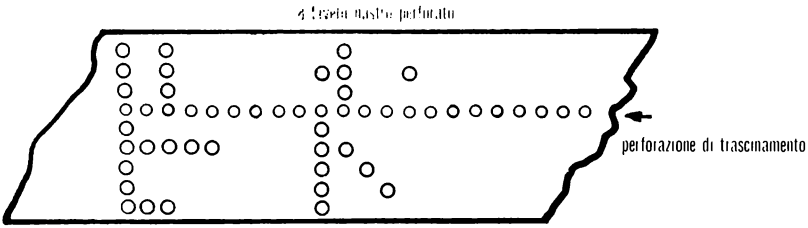


Fig. 4-40: Trama di bit a 8 livelli

nel nastro. In figura 4-41 è indicato un dato rivelato attraverso il foro. A causa di questo, il controllo del foro di avanzamento richiede un campo di variabilità in modo che il centro del foro di avanzamento corrisponda ai tempi nei quali gli altri fori sono campionati. Per far questo occorre conoscere la velocità del motore. Alcuni sistemi vanno avanti e indietro in modo che blocchi di dati con errore possano essere letti nuovamente.

In figura 4-42 è indicato il diagramma di flusso di lettori del tipo in esame.

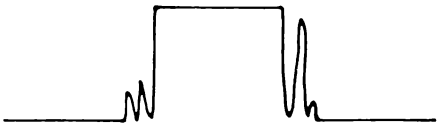


Fig. 4-41: Dato da foro

MOTORI PASSO PASSO

I motori passo passo sono un metodo comune per comandare il movimento in molti progetti. Ad ogni segnale di comando al motore passo passo, il suo albero ruota di una ben precisa ampiezza angolare. I più comuni motori passo passo ruotano di un angolo di 7,5, 15, 45, e 90 gradi ad ogni comando. Motori a passo piccolo ruotano di 1,5 e 5 gradi. Il vantaggio di una uscita a passi discreti e la possibilità di conoscere la posizione del motore in ogni momento, attraverso il conteggio dei passi che il microcalcolatore ha inviato al motore.

La interfaccia verso un motore passo passo non è semplice e può essere molto più complessa di quella che è qui indicata. Usando la fantasia per migliorare il progetto, esso può essere sviluppato in modo aderente a quanto sarà indicato.

Il motore è composto di 4 avvolgimenti. Applicando impulsi di corrente in sequenza opportuna, il motore avanza a passi successivi. Esistono tre sequenze.

- sequenza a bassa potenza
- sequenza normale
- sequenza a mezzo passo

La sequenza a bassa potenza invia un impulso di corrente in sequenza a 1, a 2, a 3, a 4, e infine torna indietro ad 1. Non sono eccitati due avvolgimenti contemporaneamente, e il motore avanza di un passo.

La sequenza normale attiva due avvolgimenti alla volta nella sequenza di seguito indicata: 1 e 2, 2 e 3, 3 e 4, 4 e 1, e così via. Questo comporta un'operazione più precisa, ma richiede più potenza.

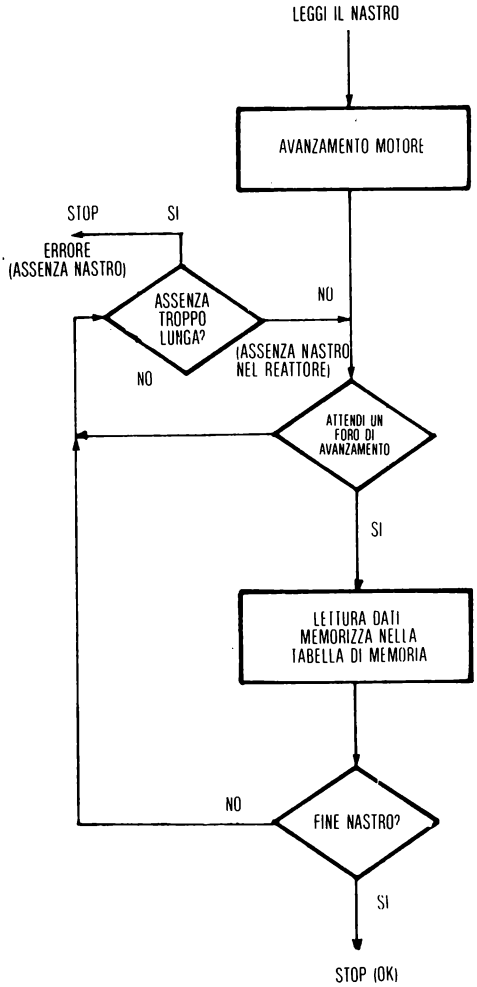


Fig. 4-42: Diagramma di flusso del lettore

Il modo a mezzo passo permette un avanzamento a mezzo angolo alla volta. La sequenza è: 1 e 2, 2, 2 e 3, 3, 3 e 4, 4 e 1, 1.

Gli avvolgimenti dei motori richiedono una sorgente di corrente poichè la loro resistenza è alquanto bassa, tipicamente 0,2 ohm. A causa, inoltre, della elevata induttanza degli avvolgimenti, sono richieste tecniche di progetto speciali per prevenire i guasti di transistori o di capacità di filtro, a causa di impulsi per scarico di e-

nergia accumulata. La figura 4-43 indica un circuito di interfaccia per un motore passo passo.

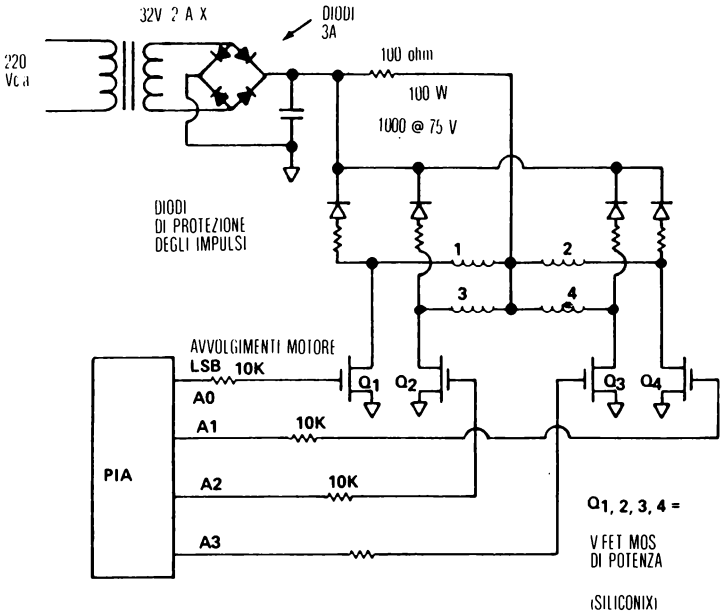


Fig. 4-43: Interfaccia del motore passo passo

La temporizzazione per il modo 2 è indicata in figura 4-44.

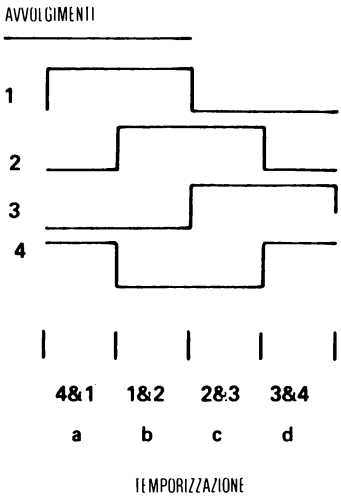


Fig. 4-44: Temporizzazione sugli avvolgimenti

È da notare che nella tabella della verità si fa scorrere mezzo byte di una coppia di 0 e di 1. Il programma è pertanto abbastanza semplice. Il programma ha il compito di ruotare ad anello il valore 00110011, presentandolo in uscita prima di ogni rotazione, verso il PIA.

TEMPO	A0	A1	A2	A3
a	1	0	0	1
b	1	1	0	0
c	0	1	1	0
d	0	0	1	1

Fig. 4-45: Tabella della verità

Nel codice del 6800 é necessario ruotare 8 bit. Poiché il 6800 ruota soltanto 9 bit, é necessario risolvere questo problema. La soluzione consiste in un semplice trucco di programmazione. Invece di ruotare é possibile aggiungere il valore a sé stesso due volte.

La sola cosa da ricordare é di aggiungere con il riporto in modo che il byte esegua un'effettiva rotazione.

```
LOOP    LDA  @ $CC        Carica l'accumulatore A con 11001100
        STA  @ $8000      Scarica l'accumulatore sul PIA
        ADC A             Aggiunge ACC A a ACC A due volte
        ADC A
        ISR delay         Ritarda il tempo dell'impulso di passo
        BRA LOOP          Vai indietro e riesegui l'anello
```

Per procedere in senso inverso é necessaria un'altra subroutine che contiene due istruzioni SBC A e due ADC A. Per variare la velocità si cambia l'ammontare del tempo perduto nella subroutine di ritardo. Come regola generale, più frequenti sono i passi del motore, maggiore é la tensione necessaria. Se si tenta di procedere troppo velocemente, il motore perde un passo. Questo é un inconveniente serio, perché non é più possibile conoscere l'esatta posizione del motore. Perdere il passo significa che il software non conosce la sua posizione. Molti meccanismi entro il motore passo passo contengono pertanto un commutatore limitatore che indica al controllore che é stato raggiunto un livello di riferimento. Tutte le misure sono pertanto eseguite facendo riferimento a questo livello di riferimento. Questo é il motivo per cui i dischi floppy leggono dalla traccia 0 in modo che l'interfaccia conosca la posizione dalla quale é iniziata la ricerca.

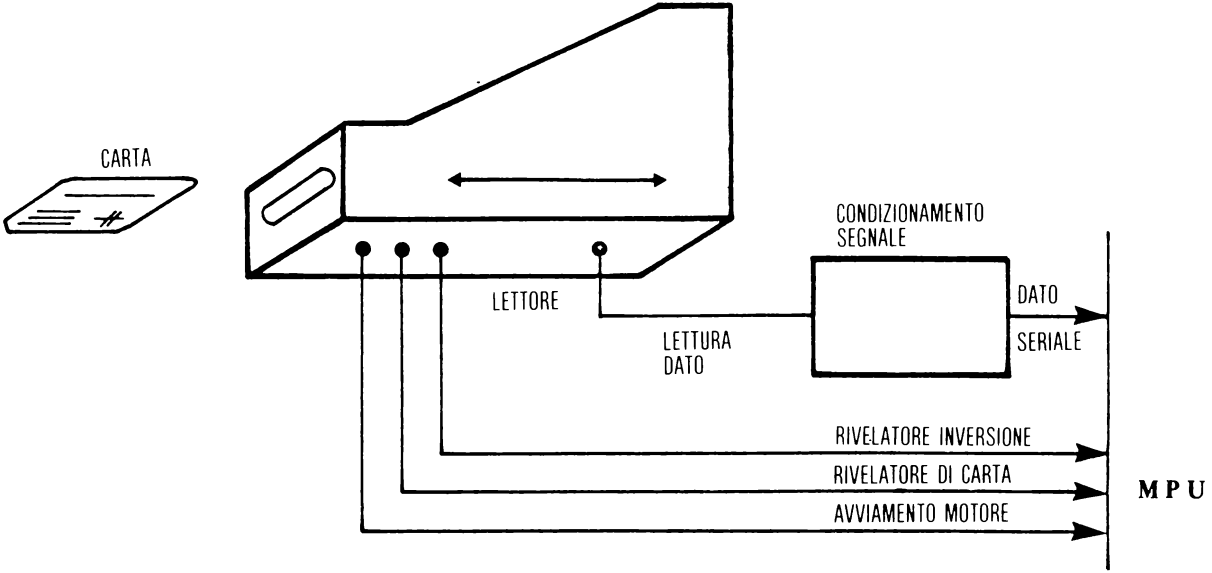


Fig. 4-46: Lettore di striscia magnetica

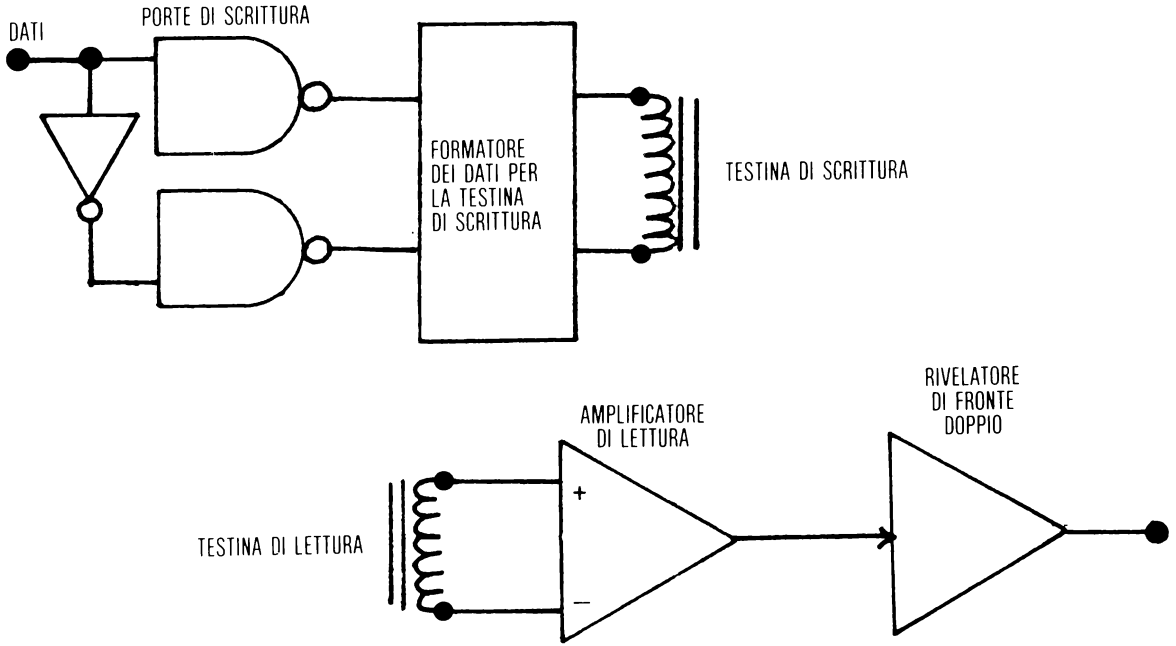


Fig. 4-47: Diagramma a blocchi del lettore di scheda con striscia magnetica

LETTORI DI CARTE DI CREDITO A STRISCIA MAGNETICA

Uno dei più recenti sviluppi nella tecnologia é stato l'uso di strisce di codifica nel retro delle bollette o nelle carte bancarie per trasferire informazioni sul conto del portatore. È di seguito descritta una interfaccia per un lettore di striscia magnetica. La interfaccia é descritta, nel suo diagramma a blocchi, in figura 4-47.

Il programma controlla la decodifica dell'informazione della striscia e il movimento della carta nel lettore. In funzionamento normale, il rullo di pressione della carta individua la sua presenza, é attivato il circuito di pilotaggio, ed é letta la striscia. Se il dato é errato o é individuata una frode, la carta é «mangiata» dal lettore. Se il dato é riconosciuto valido, é restituita la carta.

Si assume che la registrazione sulla carta sia in codice F2F, («frequenza-frequenza doppia»), nel quale un «1» corrisponde a due transizioni, uno zero «0» corrisponde ad una transizione, per ciascun bit. Pertanto i dati fuori dalla testina hanno una struttura come quella indicata nella seconda traccia della figura 4-48.

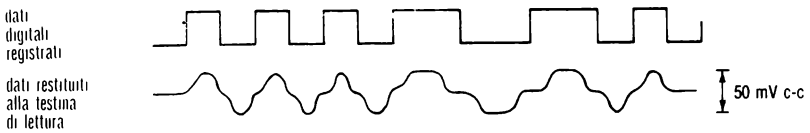


Fig. 4-48: Dati registrati

Perché tale segnale possa essere usato deve essere condizionato. Un rivelatore-amplificatore di impulsi analogici produrrà una uscita come quella di figura 4-49. Il software, attraverso cicli su una subroutine di campionamento, decodifica la forma d'onda, prima bit per bit, poi per carattere. Per assicurare dati corretti e per motivi di sicurezza i dati sono scritti *tre volte*, in forma rimescolata secondo un preciso algoritmo, con vari controlli di parità e testate, e con blocchi in coda di uni e zeri. (Vedi i formati sincroni e i codici a correzione di errore del capitolo 6).

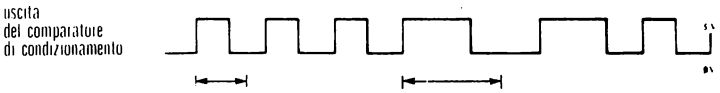


Fig. 4-49: Dati finali

Se é necessario scrivere sulle carte, si può leggere quanto già scritto e riscrivere quanto controllato. È possibile disporre di una routine software speciale che permette la rilettura prima della registrazione successiva. Il controllo necessario significa tre ingressi: rivelazione immissione carta, lettura dei dati seriali, rivelazione di

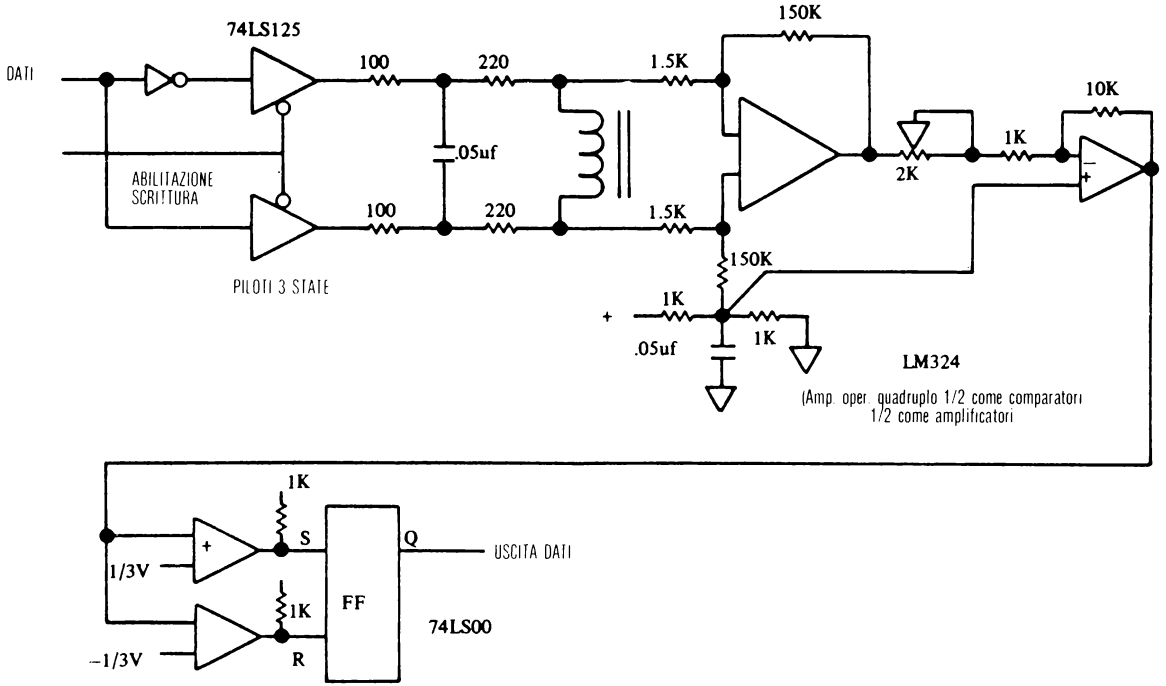


Fig. 4-50: Elettronica di scrittura/lettura

fine carta (inversione del motore per restituzione); e due uscite: attivazione del motore (l'inversione avviene automaticamente, altrimenti spegnimento) e dati seriali da scrivere. Pertanto l'hardware di uscita necessario é metà del PIA 6820 o del PPI 8255.

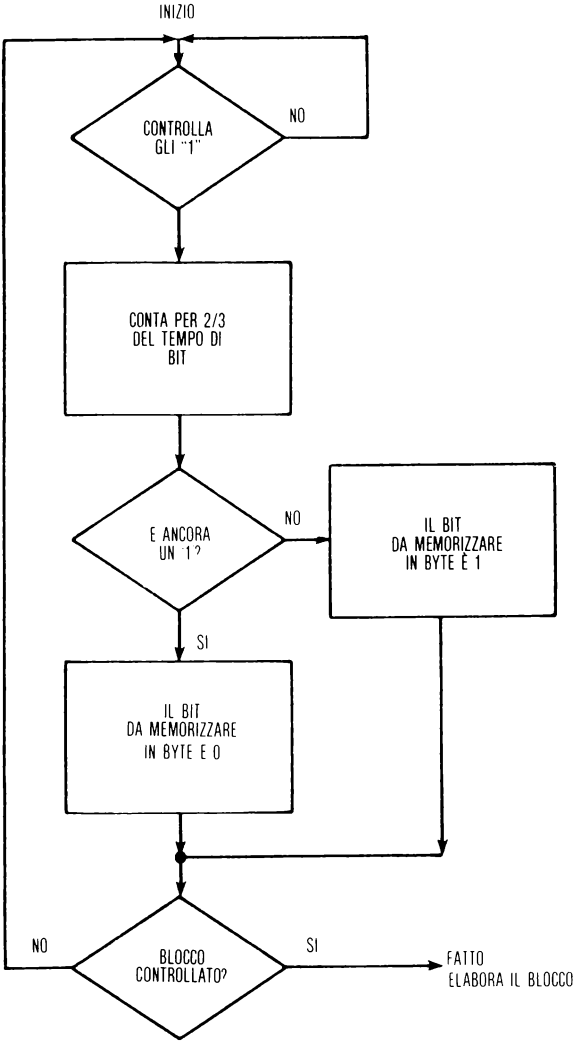


Fig. 4-51: Diagramma di flusso per un lettore di striscia magnetica

Adesso é esaminato l'hardware per scrivere e leggere sulla carta. L'elettronica di scrittura inverte la corrente delle testine di scrittura producendo un cambiamento del campo magnetico sulla striscia magnetica. In figura 4-50 sono indicati gli amplificatori di ricezione o scrittura.

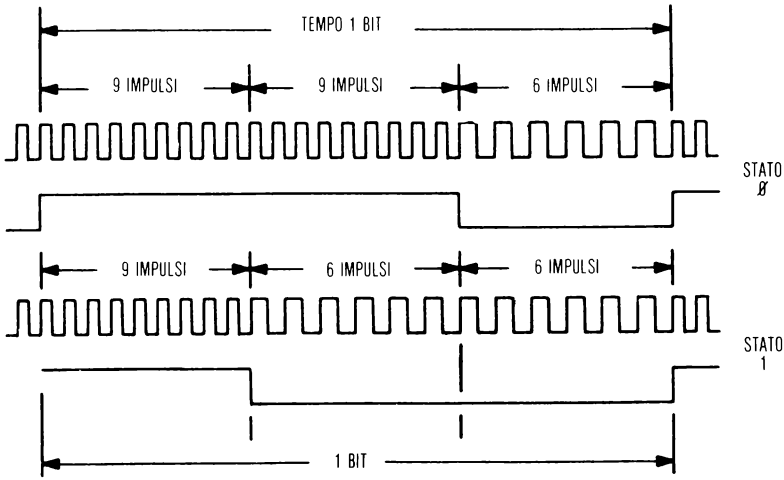
La sezione di lettura rivela sia inversioni di segnale positive che negative per assicurare rivelazione dei dati immuni da rumore. In figura 4-49 é indicata la forma d'onda d'uscita.

Il software di lettura-scrittura può essere agevolato usando uno UART o uno USART per convertire i dati, ma lo UART realizzato in software permette la massima versatilità per i formati di lettura e di scrittura. Il diagramma di flusso per la lettura e decodifica dei dati in uscita é indicato in figura 4-51.

L'INTERFACCIA DI CASSETTE KIM

Per salvare programmi e ricaricarli quando necessario, serve una forma di memorizzazione a lungo termine. Il poco costoso registratore portatile a cassette può essere usato senza modifiche per memorizzare e caricare informazioni digitali. La interfaccia richiesta é semplice da realizzare, e semplice da programmare. È qui descritta la interfaccia KIM-1® per registratori a cassette.

Il formato di trasmissione necessita di una conversione della informazione binaria in memoria in un fascio seriale di bit che può essere registrato sul nastro. Le condizioni logiche sono rappresentate da combinazioni di due toni: 3700 Hz e 2400 Hz. In figura 4-52 sono indicati i segnali relativi a un «uno» e a uno «0».



® Mos Technology Registered Trademark.

Fig. 4-52: Formato dei bit per cassetta KIM-1

Il programma genera questi toni attraverso anelli di conteggio che produrranno entrambi i toni. È usato un bit in uscita dalla interfaccia programmabile e un circuito integrato ROM presente nella piastra. Questa uscita é filtrata e tamponata mediante registro in conformità alle specifiche di ingresso di gran parte dei registratori a nastro.

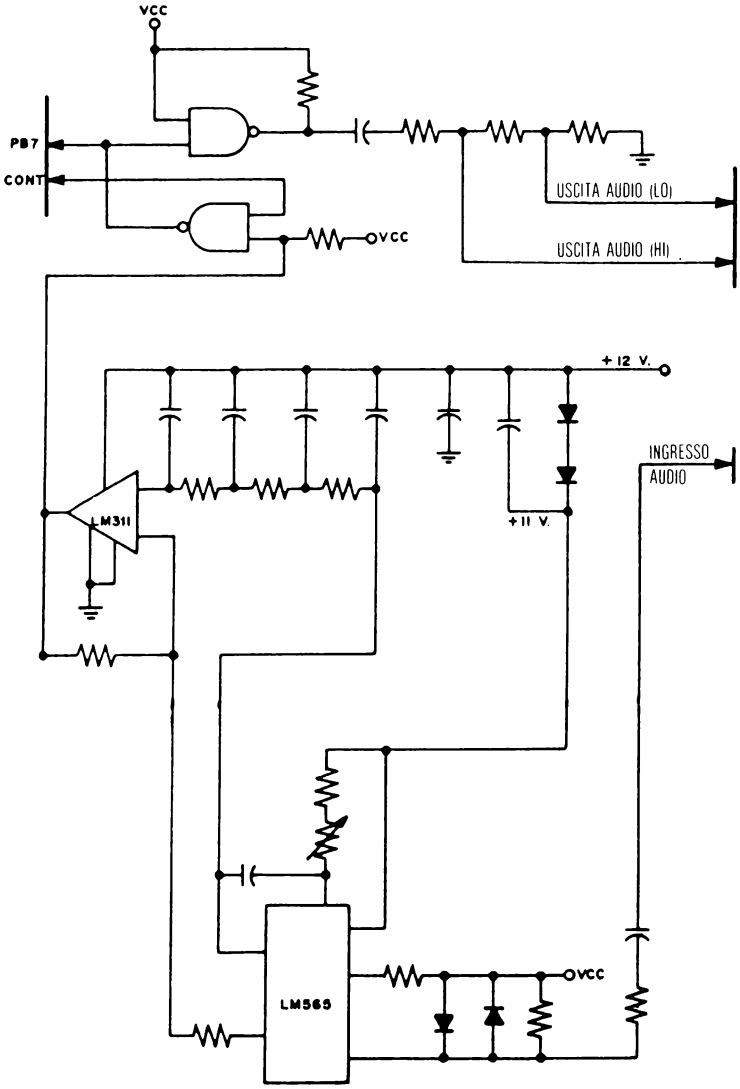


Fig. 4-53: Interfaccia verso il nastro

Quando é rivelato un tono il circuito ad agganciamento di fase della piastra riconosce un tono a 3700 Hz da uno a 2400 Hz. Campionando la durata dei toni, possono essere decodificati i bit dei dati. La figura 4-53 costituisce uno schema di interfaccia completo per il registratore a nastro.

È da notare che esistono tipi di modulazione che producono densità maggiori. Poiché tutta la temporizzazione di trasmissione e di ricezione é fatta in software, possono essere realizzati differenti metodi di temporizzazione. Tuttavia il metodo qui descritto é il più affidabile, poiché i registratori a nastro non sono adatti a densità di registrazioni più elevate, per le variazioni di velocità di scorrimento del nastro. Se sono necessarie densità maggiori deve essere usata una unità a nastro di alta qualità.

Il software divide ciascun byte di dati in due nibble di 4 bit. Ciascun nibble é successivamente convertito in un carattere ASCII a 7 bit, più la parità. Due caratteri ASCII di tale tipo adesso rappresentano il byte di dati originale. Perché il blocco originale di dati sia identificato, sono aggiunte una testata e una coda. Il formato é indicato in figura 4-54.

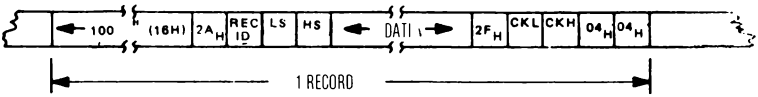


Fig. 4-54: Formati del blocco dati per il nastro

Il lungo blocco di un centinaio di 16 parole esadecimali permette al software di sincronizzarsi con la velocità di trasmissione dei dati e di trovare il primo bit di ciascun byte senza alcun'altra informazione di temporizzazione. Di seguito ai caratteri di sincronismo é presente il carattere di inizio di record, e poi il numero di caratteri del record. Sono successivamente scritti l'indirizzo di partenza del blocco dei dati, e il blocco stesso. Alla fine é scritto «2F» in esadecimale, oltre a due caratteri di controllo. Infine sono scritti due «04» in esadecimale, per indicare la fine del blocco.

Questo formato é tipico di molti schemi di trasmissione sincrona a blocchi. Altri esempi sono il disco floppy, il lettore di carte a striscia magnetica, e le vie di comunicazione tra apparecchiature (vedi il capitolo 6 per queste ultime).

STANDARD KANSAS CITY

Per usare gli economici registratori descritti nel mercato dell'hobby, é stato proposto e adottato uno standard per tale mercato. Usando tecniche di cambiamento di frequenza su chiave é semplice usare lo standard indicato mediante la tecnologia dei modem a frequenza/frequenza doppia. Inconveniente é il limite di velocità a 30 caratteri al secondo.

Il sistema usa dati seriali nello standard RS-232C (vedi capitolo 6) e converte ciascun bit o in 8 cicli a 2400 Hz (per un «1») o in 4 cicli a 1200 Hz (per uno «0»). Per ottenere ciò sono sufficienti soltanto alcuni flip-flop oltre a una porta NAND quadrupla. In figura 4-55 é indicato il modulatore.

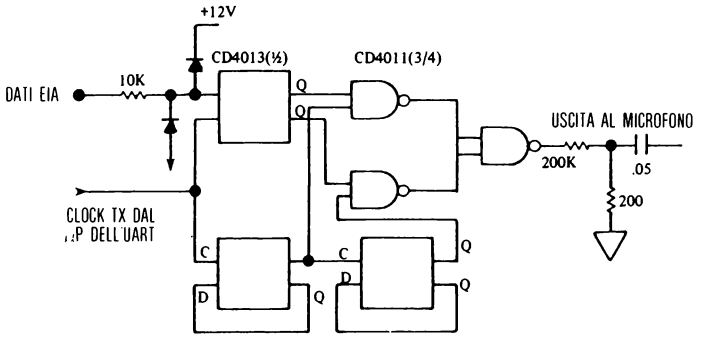


Fig. 4-55: Modulatore

Lo stato del dato di ingresso permette al multiplatore di scegliere tra i toni a 2400 Hz e a 4800 Hz. La rete di resistori é usata per ridurre l'uscita a 10 millivolt per l'ingresso da microfono del registratore.

Il demodulatore deve rivelare se è presente il tono a 1200 Hz o quello a 2400 Hz. Esistono molti modi di far questo; tuttavia quello più comune è di rivelare gli attraversamenti dello zero del segnale di ingresso. Ciò produrrà 2400 o 4800 impulsi al secondo. La decodifica è realizzata mediante un monostabile accordato in modo che se non è mantenuto «triggered» alla velocità di 4800 Hz, rivela la velocità diversa. Il vantaggio di questo metodo è di generare il «clock» necessario per lo UART del sistema, che dà ai dati la struttura corretta. L'esigenza del clock nasce dal fatto che il «clock» del ricevitore dell'UART è normalmente prelevato dal circuito indicato, piuttosto che dall'area di trasmissione.

Il circuito demodulatore é indicato in figura 4-56. In figura 4-57 é indicato il campionamento del demodulatore. È da notare che é ricostruito il segnale di partenza, oltre alla informazione necessaria per la sincronizzazione del «clock».

Se varia la velocità del nastro, il dato sarà ancora ricostruito poiché l'informazione di clock assicura che l'UART riceve l'opportuno segnale di temporizzazione. Non é necessario software speciale per questa interfaccia poichè la cassetta appare come se fosse una combinazione di un nastro perforato e di un lettore per il calcolatore.

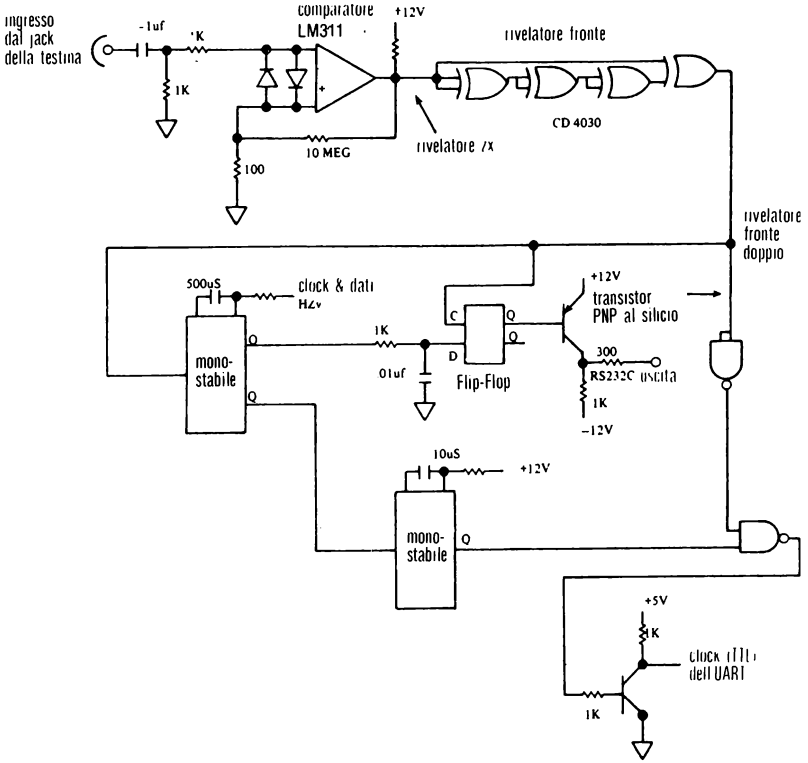


Fig. 4-56: Demodulatore

CASSETTA TARBELL

Il formato della cassetta TARBELL é una cassetta compatibile con il sistema S100 che registra a 187 Bytes al secondo o 1500 bit al secondo. La tecnica usata é quella bifase o codifica F2F come nel lettore di carte di credito. Infatti il circuito già indicato lavora altrettanto bene per realizzare una interfaccia TARBELL compatibile. Unico problema é che la circuiteria del sistema descritto realizza tutta la decodifica in software; l'unità Tarbell semplifica invece l'interfaccia disponendo di una piastra di circuiti integrati per realizzare la codifica e la decodifica del fascio di bit oltre al controllo e il pilotaggio della cassetta.

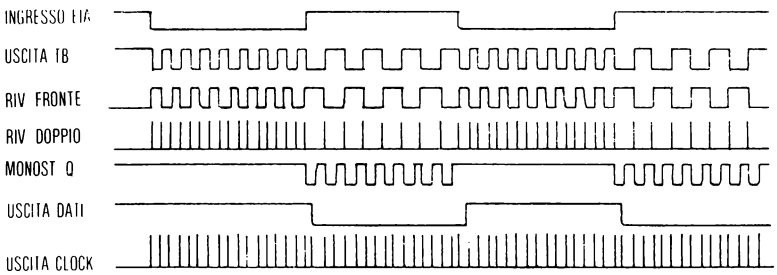


Fig. 4-57: Temporizzazione

I dati scritti sono un fascio di bit sincroni con una testata di un byte, un byte di sincronismo, i byte di dati, e byte di controllo. Una opzione é che l'interfaccia può essere regolata per leggere e scrivere anche nastri di standard Kansas City, usando una modifica software.

In conclusione questa interfaccia fornisce la circuiteria base per conversione da serie a parallelo, da dati seriali TTL a NRZ (non ritorno a zero) (vedi la interfaccia disco floppy per la sua definizione), oltre a scrivere condizionamenti del segnale per il registratore. Per la riproduzione sono disponibili il condizionatore del segnale di lettura, conversione NRZ, e conversione da serie a parallelo.

CONTROLLORE DI CASSETTA DIGITALE IN SINGOLO MODULO

Il NEC UPD371D fornisce in un unico circuito integrato gran parte delle funzioni necessarie per interfacciare una unità di trasporto di cassetta *digitale*. Esso usa il formato ISO e realizza:

- conversione di dati parallelo-serie e serie-parallelo (funzioni normalmente realizzate da uno UART)
- rivelazione di errore, includendo il CRC (il CRC sarà descritto nella sezione disco)
- formato di codifica di dati a due fasi

– essa può controllare fino a due unità di trasporto di cassetta con lettura/scrittura o riavvolgimento in una sola unità, o riavvolgimento simultaneo. Essa interfaccia direttamente lo 8080A. La struttura del sistema é illustrata in figura 4-58, mentre la sua interfaccia verso lo 8080 é indicata in figura 4 - 59.

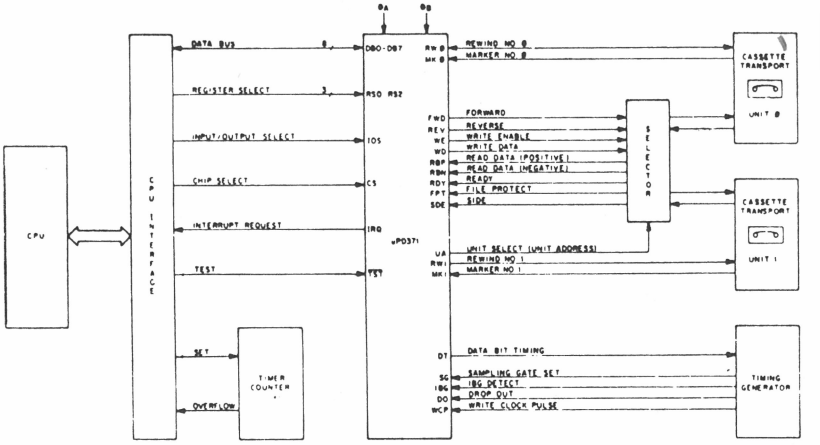


Fig. 4-58: Interfaccia di cassetta UPD 371D della NEC

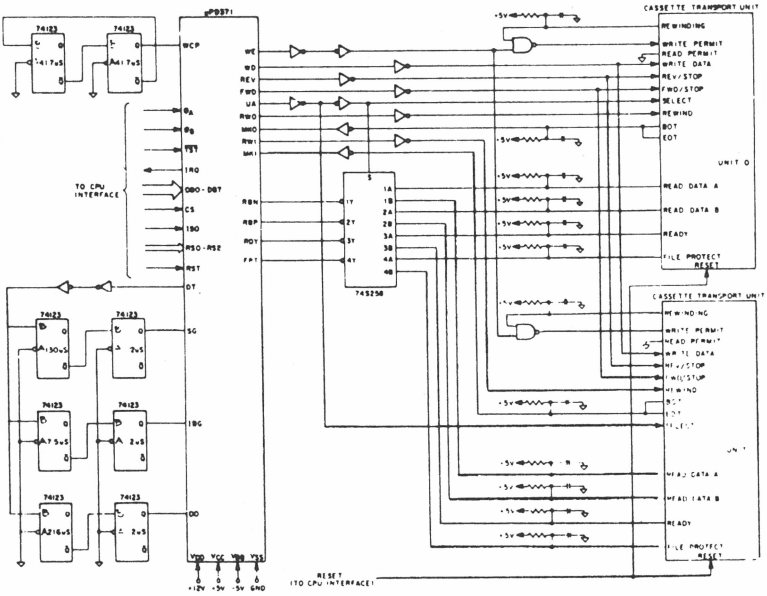


Fig. 4-59: 8080 con UPD 371D

INTERFACCIA PER TUBO A RAGGI CATODICI

Sono stati sviluppati un certo numero di CRT, da usare in modo specifico come terminali di calcolatore. Nel mondo dei microprocessori, il costo delle periferiche é di importanza critica. Pertanto il componente CRT più comunemente usato nel caso dei sistemi a microprocessore é il televisore commerciale. Schermi CRT di costo più elevato sono usati nel caso di sistemi in fase di sviluppo, per permettere all'utente di visualizzare un numero maggiore di caratteri, più linee, o più punti per carattere. Inoltre in specializzati e costosi schermi é presente una notevole capacità grafica. Sarà qui concentrata la attenzione alle interfacce per televisori commerciali.

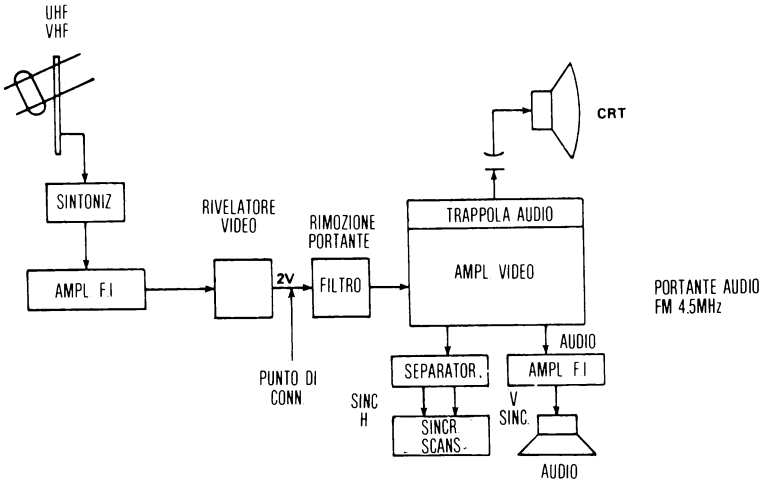


Fig. 4-60: Diagramma a blocchi di un televisore

In figura 4-60 é indicata l'organizzazione tipica di un televisore commerciale. Il segnale é fornito dall'antenna ad un circuito di accordo, che presenta in uscita una frequenza video i-f, a 4,5 MHz. Il segnale é fornito a un filtro-amplificatore che é accoppiato a trasformatore al circuito rivelatore video. L'uscita del rivelatore é il vero e proprio segnale video, variabile nel campo di 2 volt. Esso é presentato ad un filtro per rimuovere la frequenza portante e successivamente all'amplificatore video. Il segnale é infine scomposto tre volte. Il segnale video é diretto al CRT attraverso una trappola del segnale sonoro che elimina la portante del suono. La portante sonora, modulata in frequenza, é presentata in ingresso di un amplificatore del suono i-f (4,5 MHz) e l'uscita é inviata all'altoparlante.

Infine gli impulsi di sincronismo sono separati dal segnale video e identificati come sincronismo orizzontale H e verticale V. Tali segnali sono usati per sincronizzare le trame dello schermo. Il sistema microprocessore può interfacciare il video in due punti: esso può essere accoppiato direttamente all'antenna della televisione,

questo è chiamato il metodo di modulazione RF, o il segnale video può essere presentato direttamente al rivelatore video. Questo è il metodo di ingresso video diretto. Il vantaggio del metodo di modulazione RF è quello di non richiedere alcun intervento all'interno d "'apparecchiatura. I fili di uscita del microprocessore sono connessi direttamente allo schermo dell'antenna.

Oltre a richiedere conoscenza delle regolamentazioni FCC, la modulazione RF ha un problema di limitazione di banda. Usando televisori standard il limite è tra tre e 3,5 MHz. Il limite potrebbe essere apprezzabilmente minore con apparecchiature di scarsa qualità. La banda disponibile limita severamente il livello di risoluzio-

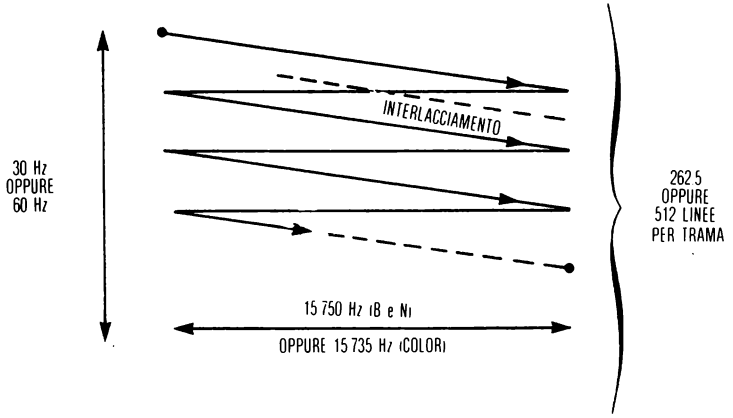


Fig. 4-61: Temporizzazione TV

ne sullo schermo e il numero di caratteri che possono essere visualizzati.

Lo svantaggio dell'ingresso video diretto è che esso naturalmente richiede una connessione all'interno del video. Poche apparecchiature dispongono di un connettore esterno connesso all'ingresso video diretto. Questo è il caso dei televisori a colori in Europa, ma non negli USA.

Per poter interfacciare verso il televisore sono di seguito descritti sommariamente i principi di funzionamento del televisore, e infine le tecniche usate per visualizzare caratteri sullo schermo.

La televisione con scansione a rastrello usa un fascio di elettroni che subisce una deflessione orizzontale di intensità variabile. Quando esso raggiunge un'estremo dello schermo è bloccato l'impatto degli elettroni e il fascio subisce una rapida deflessione sull'altro lato dello schermo, mentre è portata su un livello piú basso la scansione verticale. Questa è chiamata la fase di «fly-back orizzontale». Ciò è illustrato in figura 4-61. Sono usati due tipi di scansione, chiamati rispettivamente

scansione diretta e scansione interallacciata. Nello schema interallacciato lo schermo è scandito *due volte*. La seconda scansione, o campo, è fatta su linee intercalate alle precedenti. In ciascun campo sono disponibili 256,5 linee. Pertanto uno schema interallacciato scandisce 525 linee per trama. Nel caso di uno schermo TV connesso al microprocessore il metodo comune è non usare l'interallacciamento, e di usare una singola scansione della trama con 262 linee. La velocità di trama è pertanto 60 Hz. L'interallacciamento può essere usato per presentare titoli o messaggi. Due segnali di sincronismo sono usati per sincronizzare il movimento del punto sullo schermo con la trama: il sincronismo di linea permette il «flyback», mentre il sincronismo verticale permette il flyback sulla prima linea. Sono imposte alcune limitazioni che sono indicate in figura 4-62. La scansione orizzontale è comunemente piú larga dell'ampiezza dello schermo. L'ammontare dell'uscita dallo schermo è chiamata *sovrascansione* dello schermo. Inoltre il messaggio visualizzato sullo schermo copre un'ampiezza minore di quella consentita. Esso è indicato in figura con il termine *tempo di visualizzazione*. Quando il pennello raggiunge la fine del tempo di schermo, esso torna indietro. L'intervallo tra la fine di tempo di visualizzazione e la sincronizzazione di linea è chiamato *tempo in bianco*.

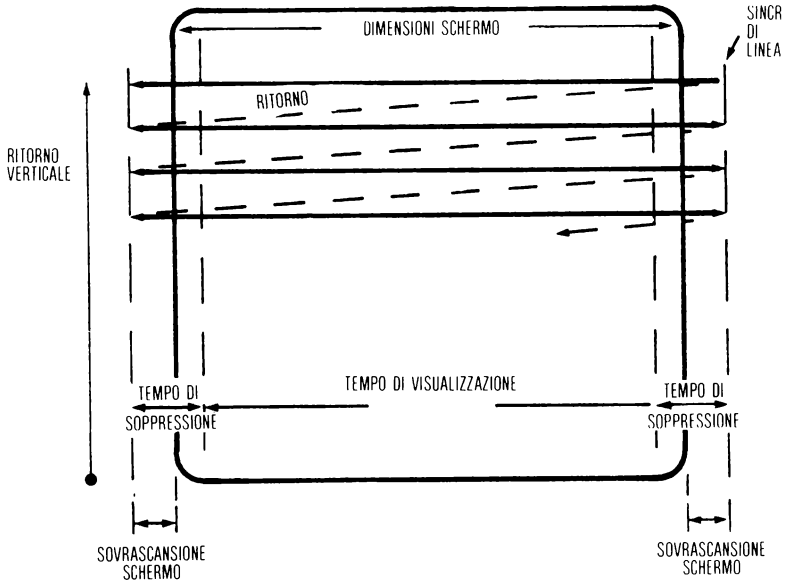


Fig. 4-62: Tempo in bianco di TV

Generazione di caratteri

I caratteri sono rappresentati sullo schermo mediante una trama di punti chiamata una *matrice di punti.* Sono usati due formati standard per rappresentare caratteri. Quella piú frequentemente usata è la *matrice di punti 5 x 7.* Un sistema meno usato è la matrice 7 x 9.

Il vantaggio di quest'ultimo è una migliore definizione dei caratteri e una piú piacevole rappresentazione di lettere minuscole. Tuttavia la rappresentazione della matrice 7 x 9 richiede l'uso di una banda maggiore e, per tal motivo, è meno usata dell'altra. Una matrice 5 x 7 rappresenta ciascun carattere con 35 punti. Essa usa 7 righe di cinque punti, e ciascun carattere è rappresentato da una sequenza di presenze e assenze di punti (punti bianchi o meglio punti in «nero»). Come i caratteri sono rappresentati è indicato in figura 4-63. Ciascuna scansione di una linea TV rappresenta sullo schermo i cinque punti relativi a tutti i caratteri della linea. Di seguito è rappresentata la riga successiva di punti di quei caratteri e cosí via. Al minimo, una matrice di punti 5 x 7 richiede otto linee dello schermo, poichè una linea in bianco deve essere lasciata fra i caratteri. In pratica, per una buona presentazione visiva, la rappresentazione di una riga di caratteri richiede dieci linee, e talvolta dodici.

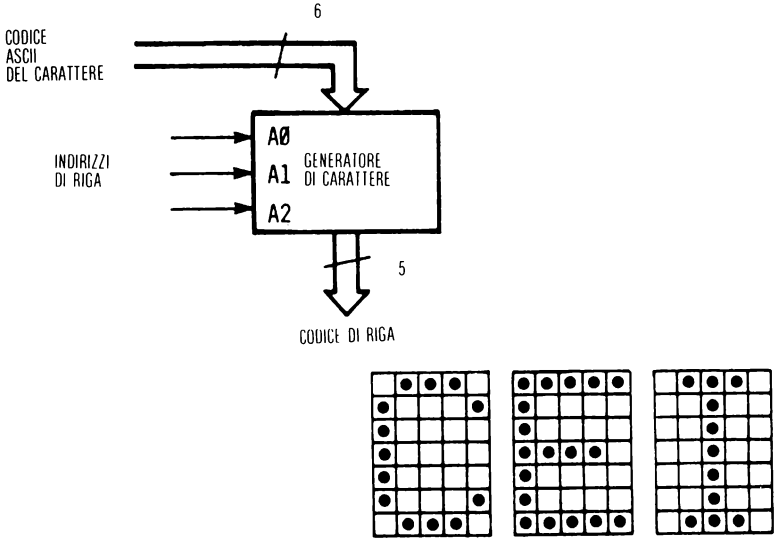


Fig. 4-63: Matrice di punti 5x7

Ciascun carattere è rappresentato entro il sistema a microprocessore mediante il suo codice, normalmente l'ASCII. La tabella di tale codice è rappresentata in figura 4-14.I 7 bit del codice ASCII devono essere convertiti in una rappresentazione di matrice a punti. Ciò può essere realizzato semplicemente attraverso un meccanismo di accesso controllato a una ROM. Può essere usato uno specifico circuito integrato o un *generatore di caratteri a matrice di punti.* Quando è usato il generatore, è presentata in sequenza la prima riga di punti di ciascun carattere di una linea: sono successivamente presentate in sequenze le altre sei righe successive. È usato un semplice contatore per tener conto delle righe di punti che sono già state prelevate dalla ROM. Nel prossimo paragrafo sarà indicato come i punti sono convertiti in segnale video da presentare all'apparecchio televisivo.

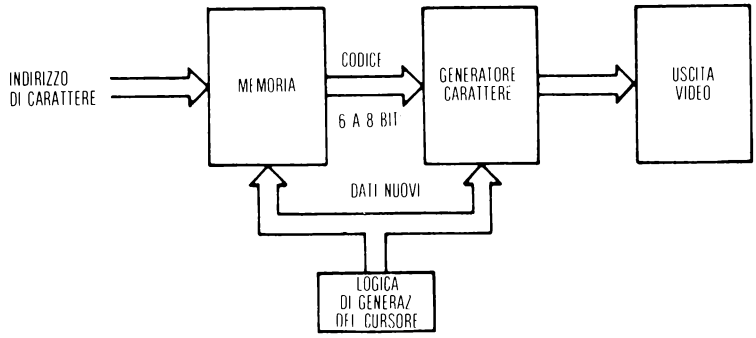


Fig. 4-64: Generazione a «raster»

Inoltre l'intero quadro, o trama, ha bisogno di essere ripetuto ad una frequenza di 60 Hz, cioè 60 volte al secondo, per evitare sfarfallìo. Ciò implica la presenza di una memoria di ripristino. La temporizzazione necessaria per il ripristino dello schermo è abbastanza veloce, da non potere usare un microprocessore standard. Circuiti esterni come il DMA, o altri circuiti speciali, dovranno essere utilizzati. Vantaggio dell'uso del DMA è che la memoria principale del microprocessore può essere condivisa, essendo utilizzata anche per il ripristino dello schermo. Tuttavia ciò riduce la velocità di funzionamento del microprocessore. In molti casi sono utilizzate *memorie dedicate* per il ripristino dello schermo. In tal caso non è ridotta la velocità del microprocessore.

Generatori di caratteri sono disponibili da gran parte dai costruttori di componenti a semiconduttore, come la Fairchild, General Instrument, Monolithic Memories, MOS Technology, American Microsystems, Electronic Arrays, Signetics e Texas Instruments.

Il numero di caratteri che può essere rappresentato sullo schermo è limitato dalla banda del terminale che è usato. Assumendo sia usato un televisore standard senza modifiche, sarà scelta una matrice a punti 5 x 7, e la combinazione piú comune è di 10 linee e di 32 caratteri, o fino a 16 linee di 32 caratteri, con un totale di 512 caratteri. Una scansione completa di linee richiede approssimativamente 63,5 microsecondi. La quota parte usabile della linea di scansione è circa 43 microsecondi. Visualizzare 32 caratteri in 43 microsecondi lascia circa 1,3 microsecondi per carattere. Ciò significa disporre di un tempo particolarmente basso per usare una memoria relativamente lenta. Se usassimo 80 caratteri per linea, sarebbe necessario un tempo di accesso di memoria minore di 0,5 microsecondi.

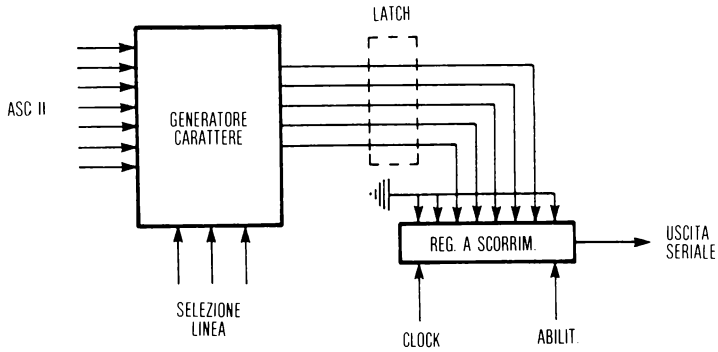


Fig. 4-65: Registro a scorrimento operante la serializzazione dei caratteri

Conversione a segnale video

I punti prodotti dal generatore di carattere devono essere opportunamente presentati in una forma seriale, per essere forniti come segnale video al televisore. Ciò è illustrato in figura 4-65. Il generatore di caratteri presenta in uscita una riga per ciascun carattere della linea. L'ASCII a 7 bit è indicato in figura sulla sinistra del generatore di caratteri, e le tre vie di selezione di linea, indicate nella parte piú bassa del generatore di caratteri, specificano quale delle sette righe della matrice di punti deve essere presentata in uscita, sulla destra. I cinque punti, corrispondenti ai contenuti di riga, sono infine caricati sul registro e temporizzati nel trasferimento in forma seriale verso l'uscita video.

Quattro tipi di dati devono essere codificati in un segnale video composito:

1. I punti rappresentanti il carattere.
2. L'eventuale segnale lampeggiante (spesso per il cursore).

3. Il cursore.
4. Infine i segnali di sincronismo H e V.

Per comporre il segnale video è normalmente usato un semplice miscelatore, con ingressi e uscite del tipo indicato in figura 4-66.

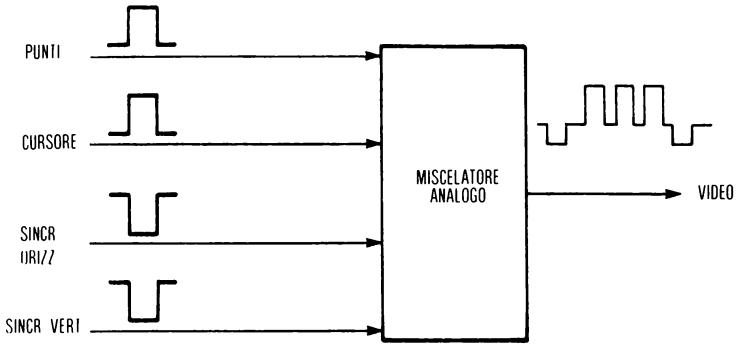


Fig. 4-66: Miscelazione per produrre il segnale video con sincronismo

I livelli di interfaccia video tipici sono da 0 a 2 volt., da 0,5 a 0,75 per il livello del nero, e da 1,5 a 2 volt per il livello del bianco, come illustrato in figura 4-67. Il segnale di sincronismo è anche chiamata la cresta di sincronismo. La sua durata è di 4,7 microsecondi. Essa è seguita dai segnali di punto in bianco e nero, codificati con un'escursione di tensione tra 0,5 volt e 2 volt. La temporizzazione è indicata in figura 4-68. In un televisore standard, il livello del bianco è al 100%, il nero è dal 25 al 30%, e il sincronismo è allo 0%. Escursione tipica di tensione è 2 volt. Il tempo standard di linea è 45 microsecondi.

Infine l'uscita video composita è connessa alla televisione o direttamente, a livello dell'ingresso video indicato, o attraverso un modulatore RF, per una connessione alla antenna televisiva. Ciò è illustrato nella figura 4-69.

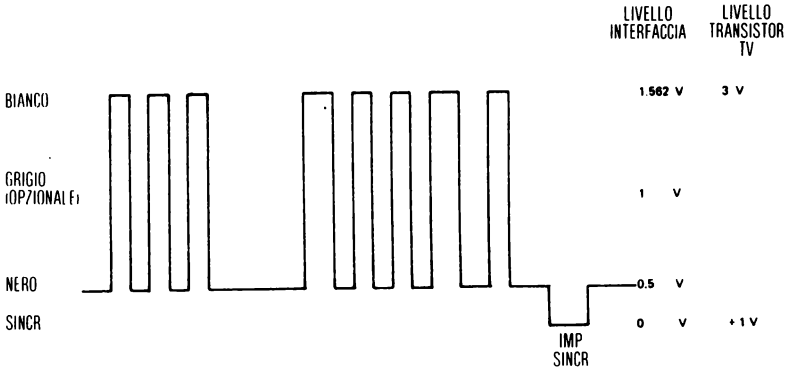


Fig. 4-67: Segnale composto da video più sincronismo

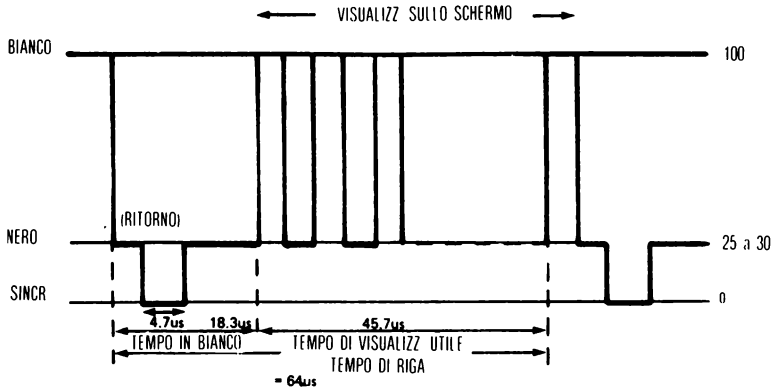


Fig. 4-68: Temporizzazione TV

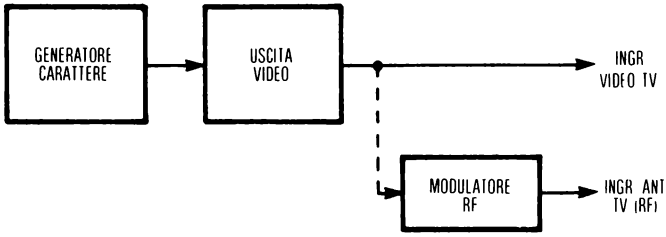


Fig. 4-69: Segnale video e ingresso RF

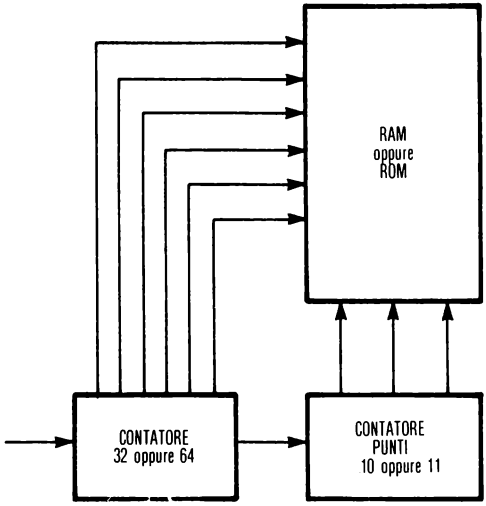


Fig. 4-70: Scansione della ROM generatrice di caratteri

Memoria di ripristino

Per semplicità di progetto, il ripristino è realizzato normalmente da una memoria dedicata. Tuttavia, un sistema a microprocessore equipaggiato con un DMA può anche essere usato per il ripristino dello schermo. In tal caso sono usate due memorie di riga durante il trasferimento DMA tra la memoria del microprocessore e lo schermo del televisore. Ciò è illustrato in figura 4-71. Il DMA riempirà inizialmente la memoria tampone 1 di linea. Durante tale tempo, la memoria tampone due di linea, che si presume piena, si svuoterà sulla via di uscita, sulla destra nella illustrazione. Tipicamente la memoria tampone due di linea si svuoterà durante un tempo 2T o piú, dove T è il tempo necessario perchè il DMA riempia una delle memorie tampone. Ogni volta che la memoria tampone due di linea si è svuotata, la memoria tampone 1 di linea, che è stata fino ad ora piena, sarà commutata e inizierà a svuotarsi attraverso il multiplatore. Non appena la memoria tampone 1 di linea è commutata, il DMA riempirà rapidamente la memoria tampone due di linea. Questo schema duale di bufferizzazione garantisce un funzionamento continuo del sistema. La sola esigenza di temporizzazione è che il DMA sia capace di riempire una delle memorie tampone di linea in un tempo minore dello svuotamento dell'altra. Chiaramente il DMA deve fare piú che questo. Il DMA deve essere capace di caricare una delle memorie tampone di linea molto piú rapidamente dello svuotamento dell'altra. Altrimenti la memoria e il DMA sarebbero praticamente usati soltanto per il ripristino di memoria, e nessun programma potrebbe essere eseguito dal microprocessore.

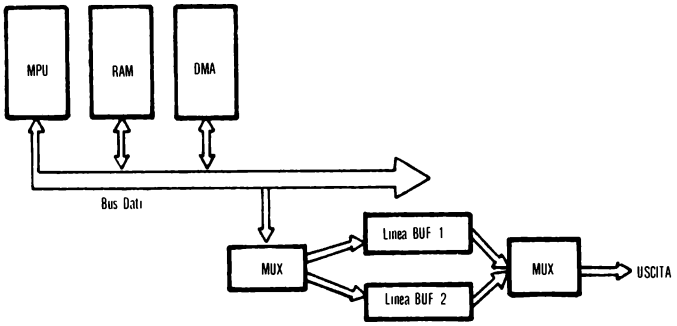


Fig. 4-71: Esigenza di memorie temporanee di linea per la MPU

CONTROLLORI DI CRT IN CIRCUITO INTEGRATO SINGOLO

I nuovi controllori di CRT integrati in un singolo circuito (CRTC) semplificano l'interfacciamento di un microprocessore a un CRT. Tuttavia, diversamente dal loro nome, essi non implementano in un singolo circuito integrato tutte le funzioni richieste per interfacciare un CRT. Essi sono progettati per CRT con scansione a rastrello, e normalmente richiedono una memoria tampone RAM di uno schermo.

Questa RAM può avere una dimensione di 2K parole o piú (richiedendo pertanto 11 uscite per indirizzo o piú). Una RAM di 2K è sufficiente per 25 linee di 80 caratteri.

FUNZIONI DI UN CONTROLLORE DI CRT

Un controllore di CRT genera essenzialmente 4 gruppi di segnali:

1 - *Indirizzo di ripristino:* indirizzo del carattere che deve essere ripristinato sullo schermo.
2 - *Selezione di riga:* per ciascun carattere, devono essere visualizzate in sequenza 7 x 9 righe di punti (usando una matrice di punti 5 x 7 o 7 x 9).
3 - *Temporizzazione della scansione video:* opportuni segnali di scansione orizzontale e verticale sono generati dalla unità (HSYNC e SYNC).
4 - *Abilitazione della visualizzazione.*

Altre due funzioni comunemente assegnate al CRTC sono:

1 - *Uscita per il cursore:* il cursore è un puntatore di carattere indipendente, una sottolineatura, una parentesi, o anche un cambiamento di colore, che può essere spostato sullo schermo, controllato da un tasto o comando speciale.
2 - *Ingresso per penna luminosa:* la penna luminosa è usata come comodo componente di accesso. Essa è sensibile al passaggio del pennello luminoso. La relazione di temporizzazione rispetto all'inizio della trama permette il calcolo della sua posizione approssimata nello schermo.

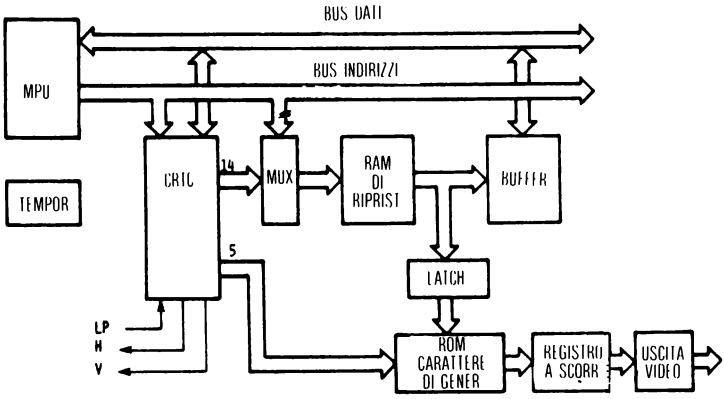


Fig. 4-72: Diagramma a blocchi di controllore di CRT

Il CRTC fornisce la logica per il controllo del cursore, la generazione dell'impulso di sincronismo, e la selezione della riga di punti in un generatore di caratteri esterno. Tutti i CRT attuali richiedono un circuito di ripristino esterno, un generatore di caratteri ROM, e la logica che sta a valle di quanto è stato descritto, includendo essenzialmente il registro di scorrimento e l'uscita video. In figura 4-72 è indicato l'uso di un CRTC del tipo descritto.

CONTROLLORE DI CRT MOTOROLA 6845

Le terminazioni in uscita dal circuito integrato sono indicate in figura 4-73. Esso genera il conteggio di riga del generatore di caratteri, il sincronismo V e H, il segnale di ritorno riga, e 14 bit di indirizzo di ripristino per la memoria tampone RAM. Inoltre esso fornisce lo *scorrimento verticale* e l'*impaginazione*. Lo scorrimento verticale consiste nello scorrimento verticale delle linee dello schermo. L'impaginazione consiste nella visualizzazione automatica di un nuovo intero schermo pieno di caratteri. Esso dispone di un registro del cursore, di un registro della penna luminosa, e non richiede una memoria tampone di linea.
Disponibilità opzionali sono:

- scansione dei punti di un singolo carattere alla volta
- numero di caratteri per linea
- numero di linee per sincronismo
- posizione del sincronismo verticale/orizzontale
- luminosità del cursore

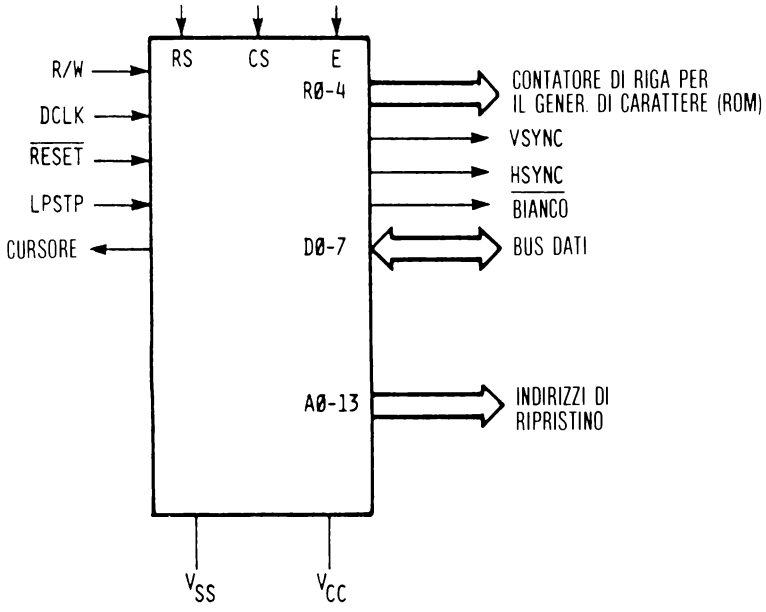


Fig. 4-73: Terminazioni del modulo per CRT

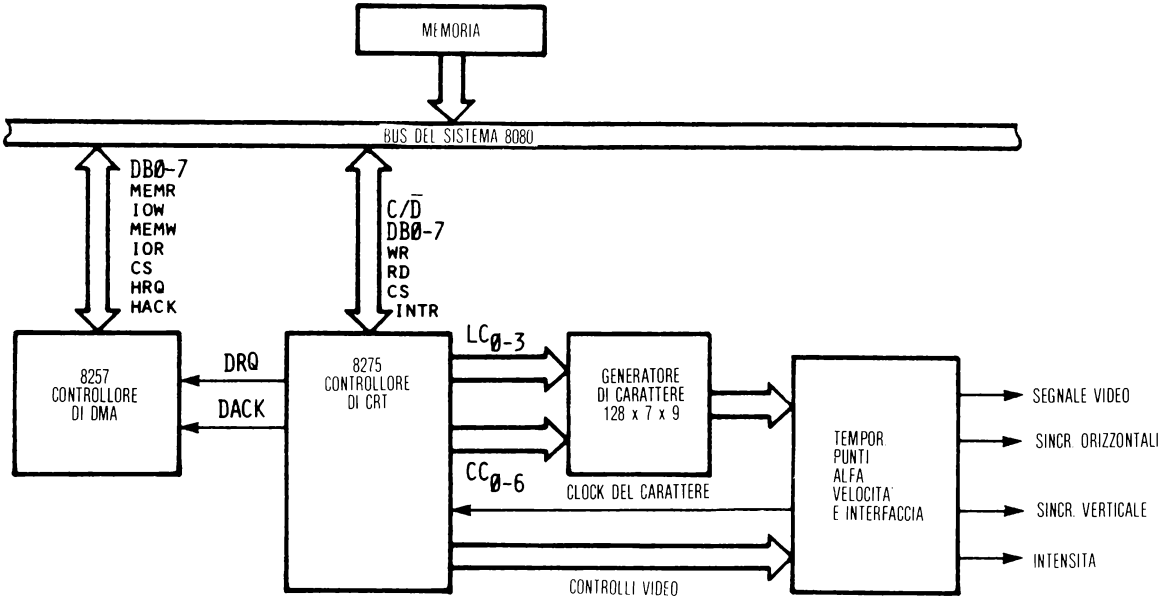


Fig. 4-74: Sistema CRTC della Intel

CRTC 8275 INTEL

In modo simile, il CRTC 8275 della Intel interfaccia un generatore di caratteri 5x7 o 7x9, e genera tutti i normali controlli video. In figura 4-74 sono indicate tutte le interconnessioni fondamentali del circuito integrato in un sistema tipico.

CRTC 9412 FAIRCHILD

Come consuetudine, il CRTC presenta 11 vie di indirizzo per indirizzare la memoria tampone. Esso include la logica per il controllo del cursore, (CM0/CM2 nella figura 4-75) e la generazione di impulsi di sincronismo (COMP SYNC, VRT SYNC).
Esso è programmabile:

- formato dello schermo (ingressi di controllo FS0-FS2)
- dimensione della matrice (5x7 o 7x9 a punti)
- auto-scorrimento su una nuova linea
- frequenza di ripristino (50 Hz/60 Hz - ingresso RR)

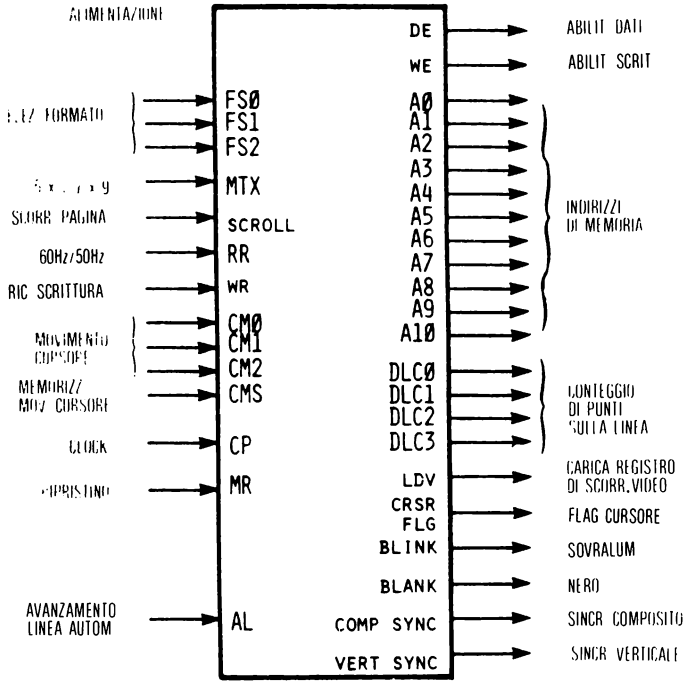


Fig. 4-75: Integrato 9412 CRTC della Fairchild

Altri segnali di uscita sono:

- DLCO-3 è il contatore di righe di punti: esso fornisce l'indirizzo della riga di ciascun carattere.
- LDV indica «video caricato». Esso è il punto di uscita verso il registro a scorrimento esterno.
- «Blank» è il segnale di ritorno riga.
- «Blink» è per fare lampeggiare il cursore o qualunque altro simbolo sullo schermo.

Come esempio, in figura 4-76 sono indicate le 8 combinazioni di codice permesse da CM0-CM1-CM2.

CRT INTELLIGENTI

Adesso che l'interfaccia hardware di un microcalcolatore verso un sistema CRT é stata descritta, é presa in considerazione la interfaccia software. Nel caso di un CRT si presenta la prima opportunità di progettare quanto considerato una interfaccia intelligente. Il termine *intelligente* indica che la periferica stessa, piuttosto che la CPU, esegue gran parte della elaborazione. Funzioni aggiunte sono la composizione, l'impaginazione e una capacità grafica. Esse sono caratteristiche desiderabili (l'ultima anche di entità limitata) in un controllore di CRT intelligente.

CM2	CM1	CM0	FUNZIONE
L	L	L	in su
L	L	H	a inizio riga
L	H	L	a sinistra
L	H	H	a inizio schermo
H	L	L	in giù
H	L	H	nuova linea
H	H	L	a destra
H	H	H	indirizzo cursore in uscita (l'indirizzo è valido quando l'uscita è a livello basso)

Fig. 4-76: Codici CMX del 9412

Dedicare un intero microprocessore a realizzare le funzioni di interfaccia di CRT, con l'aiuto di un controllore di CRT, permette di realizzare tutte le funzioni a basso costo. Terminali tipici non intelligenti realizzano opzioni di ingresso/uscita verso il sistema remoto mediante la tastiera e il CRT. Nel caso considerato, invece, le nuove opzioni possono essere aggiunte quando richieste. Dopo aver battuto uno o due paragrafi sullo schermo del nostro terminale, é possibile indicare l'errore o cambiare il testo. Mediante l'uso dei *controlli del cursore*, o la penna luminosa, o

comandi di tastiera, può essere realizzata la manipolazione dell'informazione dello schermo in modo che lo sviluppo del testo può essere semplificato. Il microprocessore riceve i comandi dal sensore di ingresso, per esempio, una penna luminosa, e riorganizza i caratteri nella memoria in modo che essi possono essere visualizzati nell'ordine nel quale l'operatore desidera siano visualizzati. È inoltre possibile aggiungere facilmente molte procedure di composizione più avanzate rispetto a quella descritta, come lo spostamento di blocchi, ricerca di stringhe di simboli, e la realizzazione di formato. Tutte queste funzioni richiedono un piccolo overhead nel programma. Tuttavia, se si considera la presenza di altri terminali intelligenti nell'intera lista di terminali, si realizza una riduzione globale di overhead.

Uno schermo CRT può normalmente contenere non più di 24 linee di 80 caratteri ciascuna; una desiderabile caratteristica é pertanto lo scorrimento e la divisione in pagine dello schermo. Questa non é una funzione difficile da realizzare poiché con un discreto ammontare di memoria un controllore può normalmente memorizzare da quattro a dieci pagine di testo dattiloscritto. Queste funzioni potrebbero essere fornite in hardware, tuttavia, attraverso l'uso di un algoritmo software, le caratteristiche di gestione a pagine del nostro terminale sono rese pii efficienti. La implementazione di comandi che muovono il testo di un certo numero di linee dentro e fuori dalla memoria ed entro lo schermo, semplificano il processo di composizione e di lettura.

Oltre alla normale presentazione del testo, é reso più semplice un certo livello di capacità grafica. Il microprocessore permetterà alle interfacce di realizzare funzioni come grafici e figure usando, per esempio, i caratteri — e +. Per esempio si può immaginare il sistema nel quale sono specificati il punto di inizio e di fine di una linea, e il terminale sceglie la migliore traiettoria per congiungere i due punti nello schermo. Tutte le caratteristiche indicate sono presenti in diverso grado nei nuovi terminali intelligenti in produzione. Esistono inoltre alcuni prodotti di tipo «personal home computing» come il Commodore Business Machines, il calcolatore PET, che presentano la possibilità di realizzare molte delle funzioni intelligenti indicate. La filosofia generale é: «una volta che io ho il microprocessore, come posso farlo operare per migliorare le funzioni di interfaccia».

CONTROLLORE DI CRT 96364 DELLA THOMSON - CSF

In Francia, la progettazione del nuovo circuito integrato di controllo del CRT ha puntato verso la riduzione del numero di interfacce.

Il CRTC 96364 può interfacciare con 19 altri circuiti SSI e MSI per fornire un'interfaccia RS232C ad un terminale CRT ASCII compatibile. È anche possibile una diretta connessione ad una tastiera ASCII. In tal modo essa rappresenta l'interfaccia di terminale a più basso costo.

Il circuito integrato CRT di base contiene i circuiti di temporizzazione e sincronismo per il televisore, la logica del cursore, contatori di schermo, e logica di controlli di una memoria esterna di visualizzazione. Esse sono illustrate nelle Fig. 4-77.

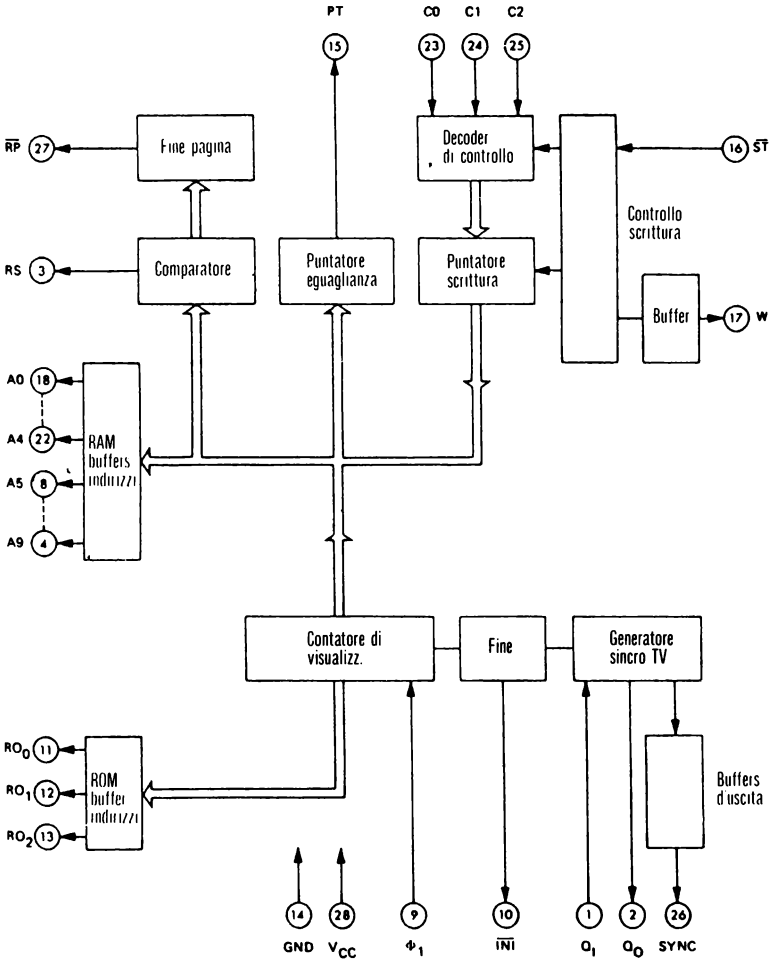


Fig. 4-77: Diagramma a blocchi del CSF 96364 CRTC

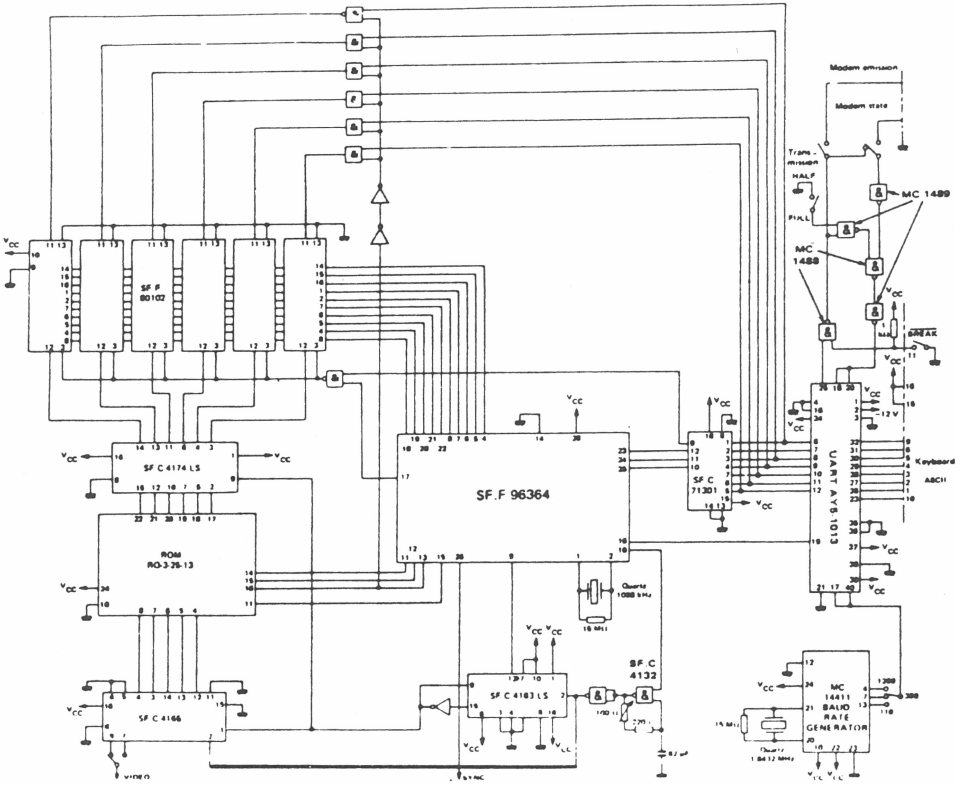


Fig. 4-77a: Terminale CRT completo

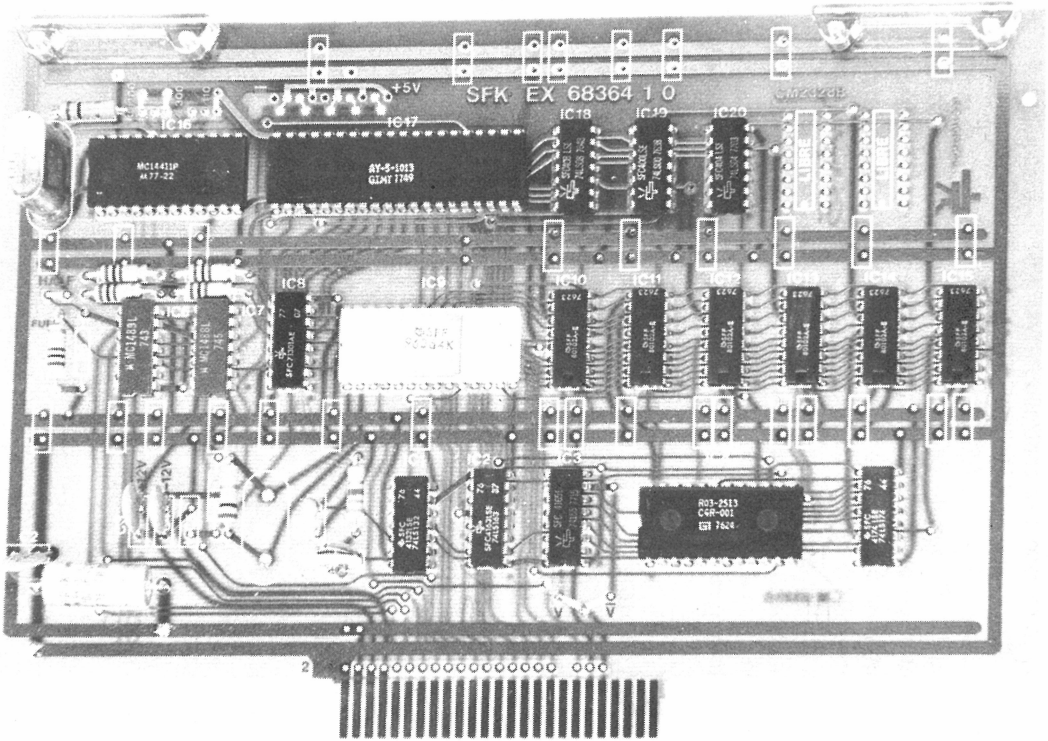


Fig. 4-77b: Piastra di interfaccia verso CRT del Thomson-CSF

In figura 4-77b il 96364 è connesso alla memoria, al generatore ROM di caratteri, a 8 registri a scorrimento seriali video, ad uno UART.

I dati relativi a caratteri entrano serialmente dalla RS232C verso l'UART. Esso converte i dati in forma parallela. I dati in parallelo sono caricati entro la memoria di schermo. La piccola ROM di controllo 32 x 4 determina se il carattere deve essere visualizzato, o se esso é di controllo: ad esempio avanzamento di linea, ritorno carrello, ecc.

Il CRTC converte i dati in ASCII in una opportuna serie di punti attraverso la ROM di carattere. Il formato del sistema é quello televisivo standard Europeo di 625 linee a 50 trame al secondo senza interallacciamento. Può essere usato un comune televisore, ma é preferibile usarne uno di tipo OEM. Inoltre gran parte del gruppo di televisori OEM sono terminali di costo molto minore di un TV comune, poiché non sono presenti alimentatori e circuiti di accordo.

Il circuito é pertanto un terminale completo con 16 linee di 64 caratteri ciascuna. I caratteri sono visualizzati in formato matrice a punti 5 x 7. Attraverso una ROM appositamente programmata (custom) é possibile rappresentare caratteri minuscoli.

INTERFACCIAMENTO DEL DISCO FLOPPY
SEZIONE I: TEORIA DI FUNZIONAMENTO

In figura 4-78 é indicato un minidisco floppy. Un *disco floppy* é semplicemente un disco codificato con un materiale magnetico, e diviso in *settori* e *tracce*, sui quali é registrato il dato. Esso fornisce un mezzo di memorizzazione a costo molto basso con accesso ad alta velocità e elevata capacità. Attualmente esistono due tipi di dischi floppy: il disco floppy regolare e quello mini.

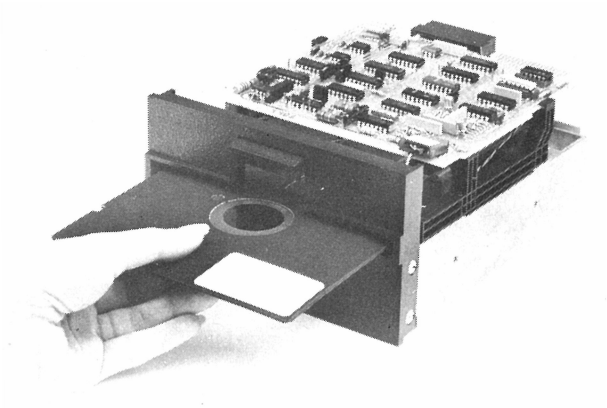


Fig. 4-78: Mini-floppy della Shugart

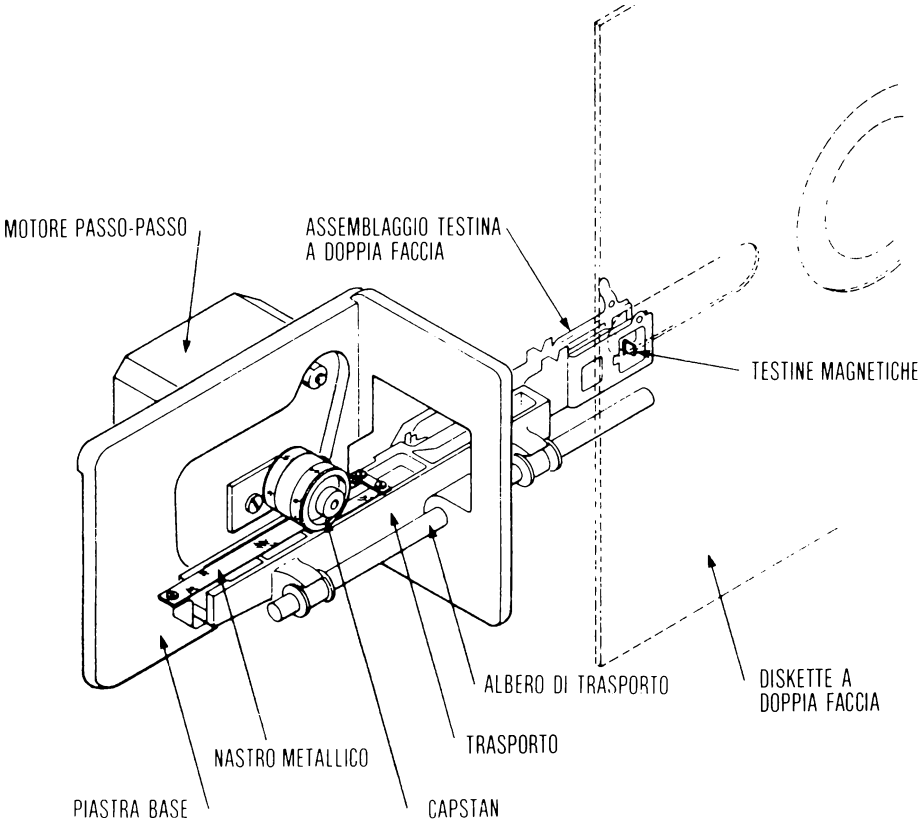


Fig. 4-78a: Dettagli del sistema di posizionamento

Un disco floppy regolare come quello SA800 SHUGART fornisce i seguenti servizi: (Esso può essere a densità singola o doppia. È qui considerata la densità singola).

– Capacità totale per disco: 3,2 Megabits
– Capacità per traccia: 41,7 Kilobits (senza formato).

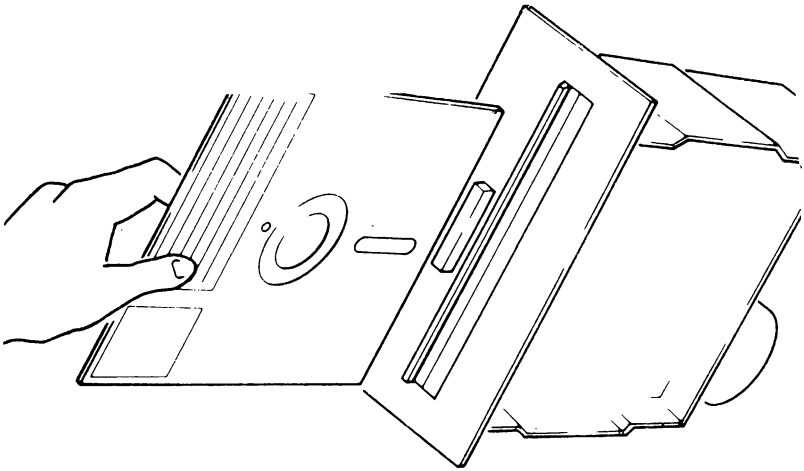


Fig. 4-79: Dischetto floppy e pilotaggio

Specifiche tipiche dei floppy

Dimensione:	dischetto 8" formato: 76 tracce + traccia indice (formato IBM 3740) 26 settori per traccia
Capacità:	128 Bytes per settore 3,3 K Bytes per traccia 253 K Bytes per dischetto
Densità:	Tracce: 48 tpi bits: 3268 bpi (densità normale). Densità doppia: 6536 bpi
Velocità:	rotazione: 360 rpm ± 2% trasferimento: 250 K bps. densità doppia: 500 K bps
Temporizzazione:	passo da traccia a traccia: da 10 a 18 ms (inclusi da 8 a 15 ms per posizionamento testina) ricerca: massimo da 100 a 768 ms blocco testina: 40 ms accesso medio: da 136 a 476 ms

Affidabilità:	(dati Persci) errore lettura (soft) minore di uno su 10^9 bit errore di lettura (hard) minore di uno su 10^{12} bit errore di posizionamento minore di uno su 10^6 accessi
MTBF:	più di 4000 ore
MTTR:	minore di 20 minuti
vita:	1500 ore o 5 anni

Per esempio, il tempo di accesso per un normale disco floppy come lo SA800 SHUGART (assumendo densità normale):

- da traccia a traccia: 8 ms
- tempo medio di accesso: 250 ms
- tempo di assestamento: 8 ms
- tempo di impegno della testina: 35 ms
- la velocità di rotazione del disco é di 360 rpm e la densità di registrazione (entro la traccia) é di 3200 bpi in densità semplice, e 6400 bpi in densità doppia
- la densità di traccia é di 48 tpi e il numero delle tracce é 77.

Specifiche del minifloppy

Dimensione:	5,25" per i minidischetti
Formato:	23 tracce
Capacità:	1/3 del floppy standard 89K bytes/minidischetto
Velocità di immagazzinamento (spool):	da tre a 6 volte minore del floppy normale

Formati entro il dischetto

Ciascun dischetto usa normalmente il formato tipo IBM 3740, con 77 tracce, numerate da 00 (la più esterna) a 76 (la più interna). Una delle tracce é usata normalmente come indice, restando 76 tracce da usare per i dati.

Ciascuna traccia é divisa in settori (come fette di una torta). Sono usate due tecniche per definire i settori: settori in hardware e settori in software.

Nella tecnica dei *settori in hardware*, 32 fori sono perforati nel dischetto, definendo 32 settori di 128 bytes. Ciò comporta la più alta densità dei dati.

Nella tecnica dei settori in software, é perforato soltanto un foro nel dischetto, indicando l'inizio del settore zero. Il numero dei settori é così definito dall'utente. Il formato IBM compatibile definisce 26 settori di 128 bytes. Poiché ciascun settore deve essere chiaramente identificato, i settori sono separati da spazi vuoti (gaps) e preceduti da una testata contenente la loro identificazione. Ciò produce una più

bassa densità dei dati che con la realizzazione dei settori in hardware. Tuttavia, ciascun settore é chiaramente identificato ogni volta che si accede ad esso, comportando una maggiore affidabilità.

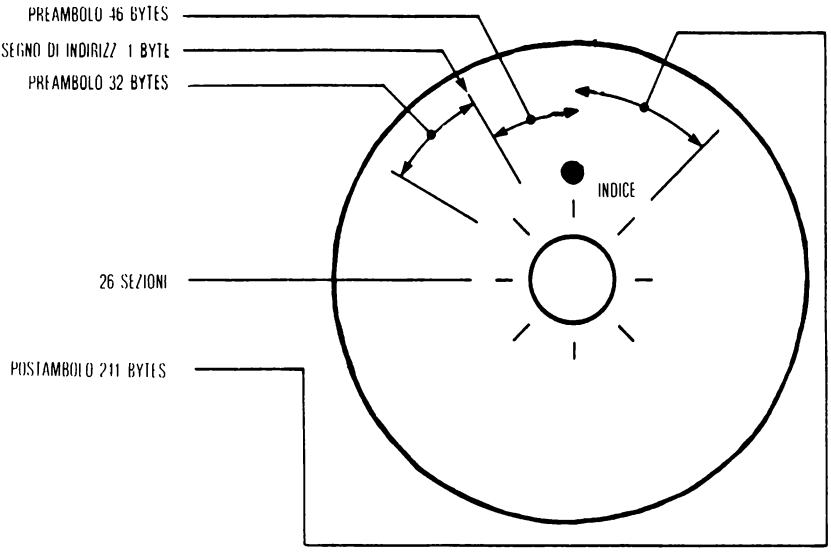


Fig. 4-80: Formato del disco floppy

Caratteristiche del mini-floppy

Le caratteristiche di un mini-floppy sono le seguenti:

- capacità:
  - senza formato: 109,4 Kbytes per disco e 3125 bytes per traccia
  - con formato: devono essere distinti due casi: formato software e formato hardware.

In *formato hardware*, i fori sono perforati nel disco, per indicare l'inizio di un nuovo settore. In formato software, soltanto un foro é perforato per indicare l'inizio di ciascuna traccia, e la lunghezza dei settori nella traccia é lasciata libera alla scelta del progettista o del programmatore.

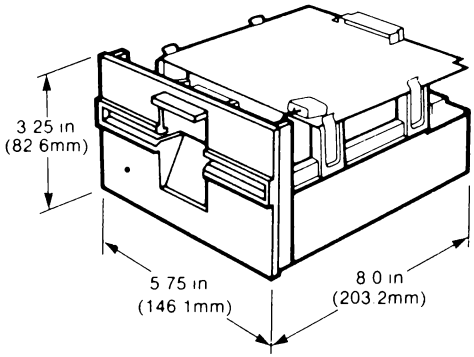


Fig. 4-81: Dimensioni di un mini-floppy

Software	**Hardware**
Per disco: 80,6 Kbytes	72,03 Kbytes
per traccia: 2304 bytes	2058 Kbytes
per settore: 128 bytes	128 bytes
settori/traccia: 18	16 Kbytes

La velocità di trasferimento é di 125 Kilobit per secondo.
Il tempo di accesso é:
da traccia a traccia: 40 ms
medio: 463 ms
Il tempo di assestamento é 10 ms
Il tempo di impegno della testina é 75 ms
La velocità di rotazione é: 300 rpm
La densità é 2581 bpi (per la traccia interna)
Il numero totale di tracce é 35
La densità di traccia é 48 pbi.

Segnali di interfaccia base per il pilotaggio del disco

I segnali di interfaccia includono comandi e dati verso l'unità di pilotaggio e stato più i dati da quest'ultima verso l'unità di controllo. In particolare, verso il pilotaggio:

- impulsi di avanzamento verso il motore della testina + comandi di direzione
- impegno testina
- lettura/scrittura
- dati + informazioni di temporizzazione
- bit(s) di reset di errore

Dal pilotaggio:

- impulso di indice
- impulsi di settore (se settori in hardware)
- bit(s) di rivelazione errore
- indicazioni di protezione alla scrittura (controllo «tab») nel dischetto
- dati + segnali di temporizzazione
- traccia 00

Il pilotaggio del disco floppy

Il circuito di pilotaggio contiene le funzioni meccaniche e la elettronica richiesta per far ruotare il dischetto e per accedere ai dati.

Il dischetto é un disco di milar flessibile («floppy») ricoperto di un ossido magnetico. Esso ruota entro un involucro. Un lungo foro é realizzato nell'involucro lungo il raggio per permettere l'accesso alla testina di scrittura/lettura.

La stessa testina é usata per lettura/scrittura e cancellazione. La testina é spostata lungo il raggio attraverso un motore di posizionamento, normalmente un motore passo passo. Una volta che essa é stata posizionata sulla traccia richiesta, la testina é posta in diretto contatto con la superficie del dischetto.

Inoltre l'involucro ha un foro indice. Il foro indice é perforato nel dischetto, e fissa l'inizio del settore zero. Esso é rivelato da un circuito fotosensibile nel circuito di pilotaggio.

L'elettronica di pilotaggio realizza 4 funzioni:

1 - Muove la testina verso la traccia
2 - Impegna la testina e compie una lettura o una scrittura
3 - Genera o interpreta i segnali di controllo o le informazioni di stato (inclusa la rivelazione dell'indice, la rivelazione della traccia zero)
4 - Pilota con precisione il motore dell'albero.

Opzioni per il pilotaggio sono:

- comando remoto di espulsione del dischetto
- protezione alla scrittura

Segnali tipici tra il FDC e la MPU

Verso la MPU

- richiesta interruzione
- richiesta trasmissione
- dati di 8 bit

Dalla MPU

- dati di 8 bit
- CLK
- RES
- R/W
- impulsi di selezione (connessione al bus di indirizzo)
- conferme di avvenuta ricezione

Funzionamento del pilotaggio del floppy

Il principio della operazione di lettura e della scrittura é di accedere alla specifica traccia e allo specifico settore, e di trasferire allora un blocco di dati. Tre operazioni sono pertanto realizzate: *posizionamento della testina, controllo scrittura/lettura,* e *trasferimento dati.*

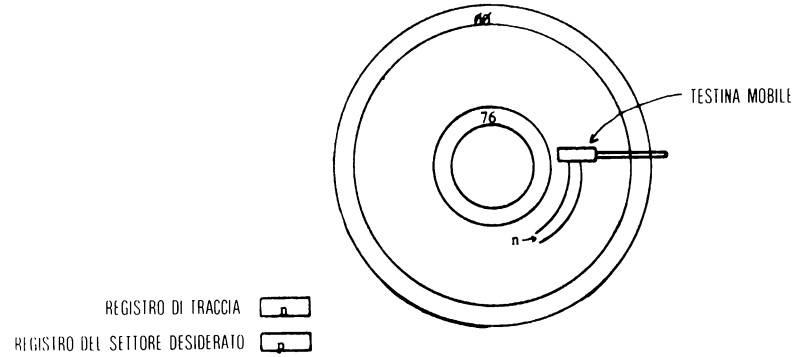


Fig. 4-82: Accesso al settore p della traccia n del generatore/controllore dei formati

1 - Posizionamento della testina:
La testina é fatta procedere a passi mediante un motore passo-passo incrementale (tipicamente da 3 a 10 ms per passo). Ciò implica la necessità di un ritardo per passo programmabile nel controllore del formato, qualunque esso sia. Naturalmente una via deve specificare la direzione del movimento. Deve essere anche previsto un ritardo di posizionamento della testina, dell'ordine di 8-15 ms, necessario per le estinzioni delle vibrazioni.
La testina é allora posta a contatto con il disco (10 ms di assestamento). È allora necessario verificare il posizionamento corretto leggendo il primo campo ID nel dischetto. Esso è confrontato con il registro di traccia. Inoltre il campo CRC dello ID é controllato per verificare l'integrità della informazione. Si può, a tal punto, procedere all'accesso.

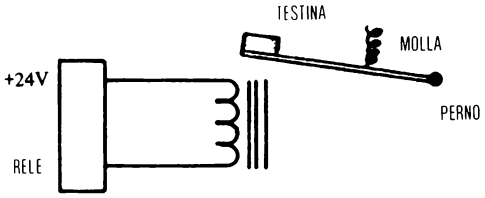


Fig. 4-83: Pilotaggio del disco: assemblaggio dell'impegno testina

2 - Controllo R/W:
Assumendo che il componente e i dati siano disponibili, é attivata la porta di scrittura (per una scrittura). L'operazione é inibita se il dischetto é protetto contro la lettura.

3 - Trasferimento dati:
Il trasferimento deve avvenire ad una velocità fissata. Una temporizzazione tipica é di un MHz in densità normale (0,5 MHz per un minidisco floppy), 2 MHz in densità doppia (1 MHz per un minidisco floppy).

Un disco può essere semplicemente protetto contro cancellazioni accidentali usando un passante di protezione alla scrittura sulla busta del contenitore del disco. Ciò è illustrato in figura 4-84.

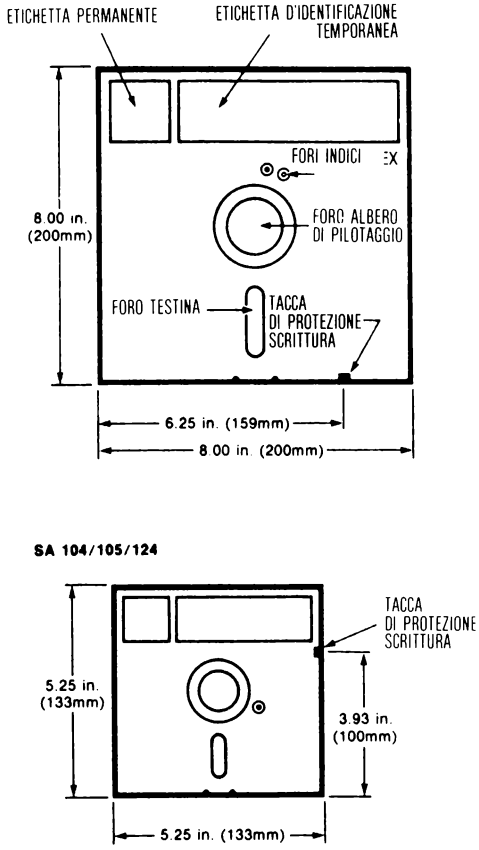


Fig. 4-84: Confronti: «floppy» e «flippy»

Il pilotaggio del disco

Il circuito di pilotaggio del disco contiene esso stesso le seguenti caratteristiche:

1 - Controllo lettura/scrittura, più la elettronica di controllo (2 piastre di circuito stampato)

2 - Il meccanismo di pilotaggio

3 - Il meccanismo di posizionamento della testina di lettura/scrittura

4 - La testina di lettura/scrittura

I servizi ausiliari di lettura/scrittura, indicati nel punto 1 precedente, includono:

- riconoscimento dell'indice e del settore
- esecutori del circuito di pilotaggio per il posizionamento della testina R/S
- esecutori del circuito di pilotaggio per il contatto con il disco R/W
- pilotaggio della scrittura
- amplificatore di lettura, più rivelatori di transizione
- riconoscimento di protezione contro la scrittura
- circuiti di pilotaggio per la selezione
- circuiti di controllo del motore

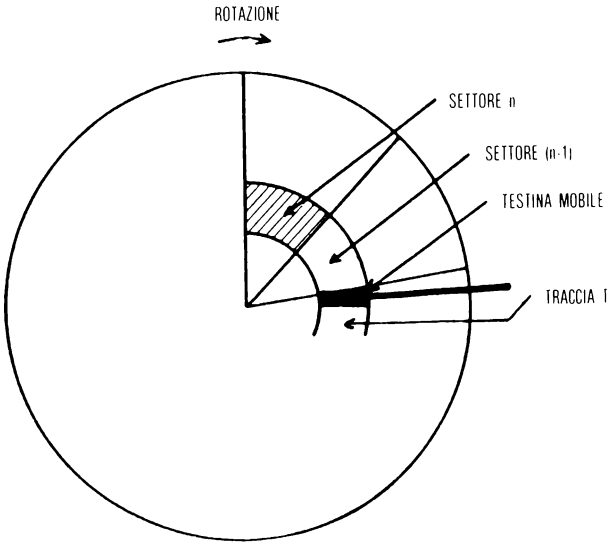


Fig. 4-85: Posizionamento della testina sulla traccia corretta

Accesso alla traccia

La testina si muove sulla superficie del disco da una traccia all'altra. Essa é mossa radialmente nel disco da un motore passo passo. Per accedere alla traccia sono eseguite le seguenti sequenze:

1 - Deve essere attivata la selezione del circuito di pilotaggio. Normalmente un controllore di disco può controllare più di un'unità, e pertanto abiliterà il selettore di pilotaggio del meccanismo che é selezionato: per l'accesso.

2 - È posizionato il selettore di direzione, risultando così fissata la direzione del movimento della testina. La testina può muoversi sia verso il centro del disco che verso la periferia.

3 - La porta di scrittura é disabilitata. Durante il movimento della testina non può essere, infatti, abilitata la scrittura.

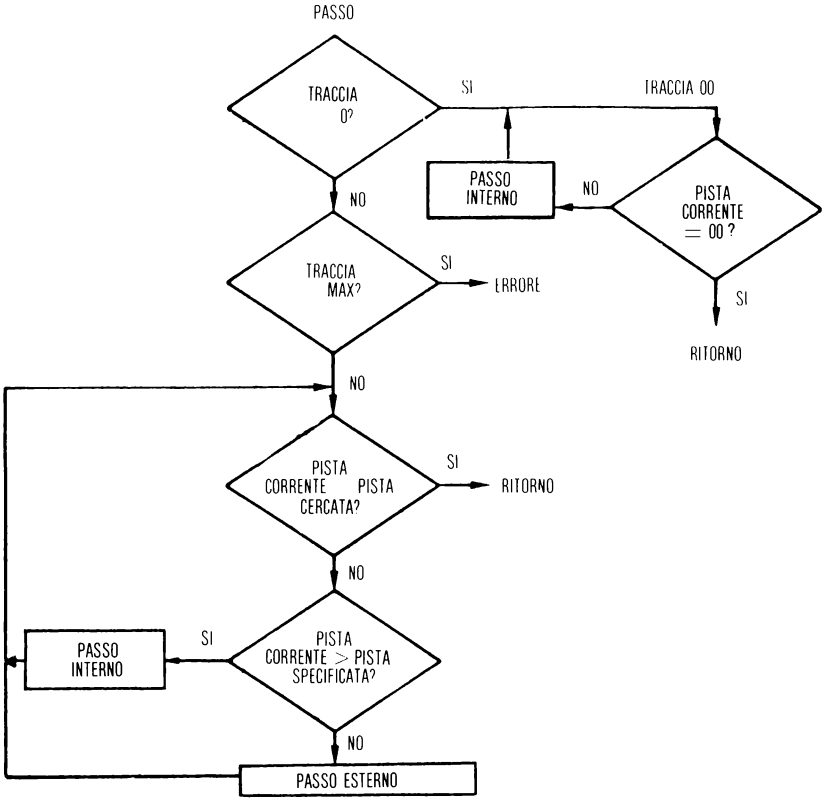


Fig. 4-86: Criterio di spostamento verso una traccia specificata

4 - Sono forniti impulsi sulla via per lo spostamento passo passo finché é raggiunta la traccia desiderata. A ciascun impulso corrisponde lo spostamento di una traccia della testina, nella direzione scelta.

Lettura e scrittura

La *lettura* é semplicemente realizzata mediante:

– attivazione del selettore di circuito di pilotaggio
– disabilitazione della porta di scrittura

La *scrittura* é realizzata mediante:

– attivazione del selettore di circuito di pilotaggio
– attivazione della porta di scrittura
– invio di impulsi di dati sulla via di scrittura dati.

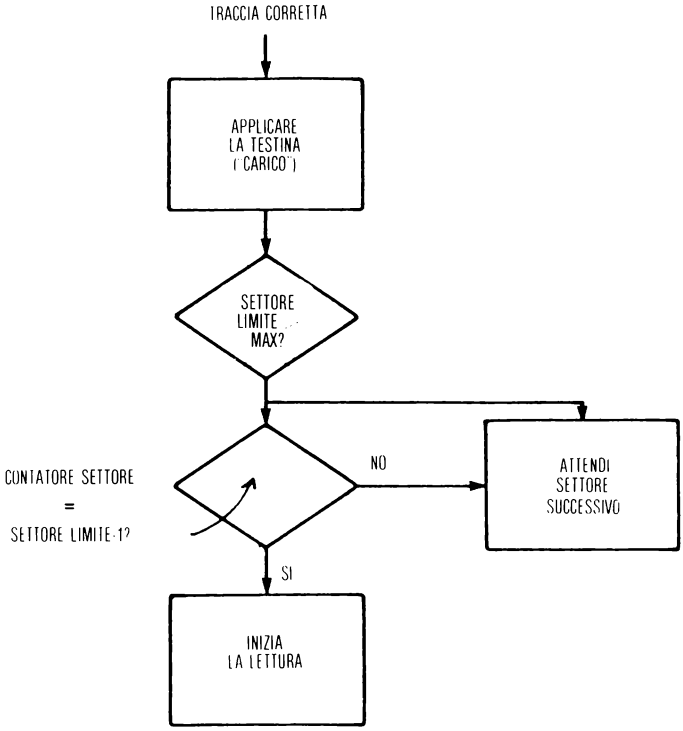


Fig. 4-87: Accesso al settore

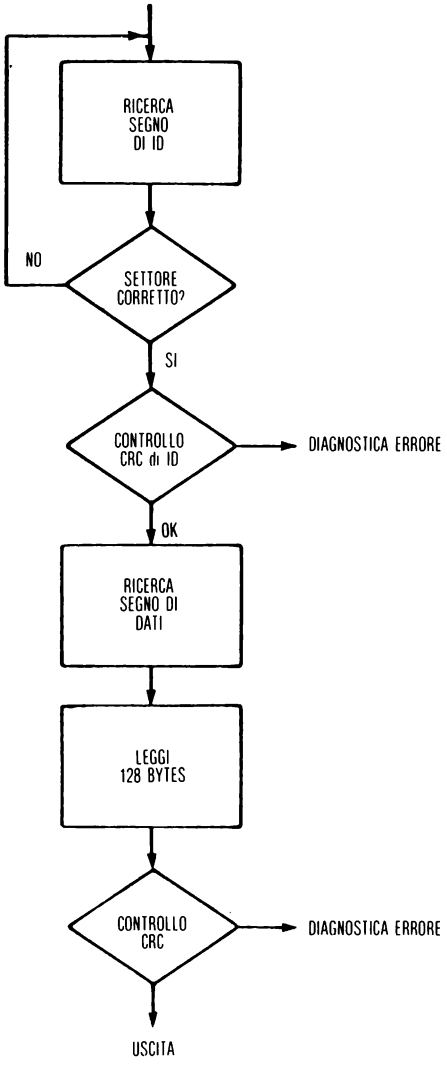


Fig. 4-88: Una sequenza reale di lettura

Segnali di pilotaggio del disco: un esempio

I segnali richiesti e generati dal circuito di pilotaggio del minidisco floppy SA 400 sono indicati in figura 4-89. Sei segnali fondamentali sono usati per comunicare con il circuito di pilotaggio del disco:

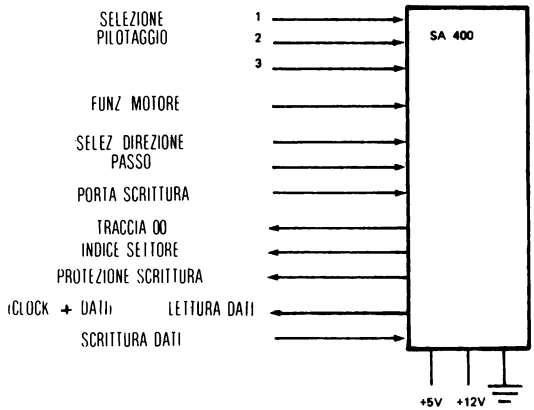


Fig. 4-89: Pilotaggio del disco floppy SA 400

ATTIVAZIONE DEL MOTORE

Il segnale attiva e blocca il motore. Quando il motore é attivato, deve trascorrere un secondo dopo l'attivazione, prima di procedere.

Corrispondentemente, il pilotaggio del disco é disattivato dopo 10 secondi (o 10 rotazioni), ogniqualvolta non é fornito alcun altro comando. Questo allunga la vita al circuito di pilotaggio.

SELEZIONE DELLA DIREZIONE

Questo ingresso seleziona la direzione nella quale la testina di lettura/scrittura é mossa. L'effettivo spostamento é realizzato fornendo impulsi alla via di comando degli spostamenti passo - passo.

PASSO

Esso muove la testina dalla posizione di una traccia a quella adiacente, verso il centro o in direzione opposta. Lo spostamento avviene sul fronte di discesa dell'impulso.

PORTA DI SCRITTURA

La scrittura é abilitata quando questa via é attiva. Quando essa é invece inattiva é specificata una lettura.

TRACCIA 00

Il segnale indica che la testina ha raggiunto l'estremo esterno del disco, cioé la traccia più esterna o traccia 0. La testina non si muove più, anche se sono forniti altri comandi di ulteriore spostamento.

INDICE SETTORE

È fornito un segnale ogni volta che é rivelato un foro nel disco. Possono essere usati due tipi di fori, fori di indice, e fori di settore. Ogni disco presenta un *foro-indice* indicante l'inizio del primo settore del disco.

Un disco con formati in hardware, che sarà descritto sotto, ha un numero addizionale di fori indicanti l'inizio di ciascun settore. Quando sono usati formati in software, é fornito un impulso per ciascuna rotazione, all'inizio di una traccia. Ciò avviene ogni 200 ms. Quando i settori sono indicati in hardware, 11 o 17 impulsi sono forniti per ciascuna rotazione.

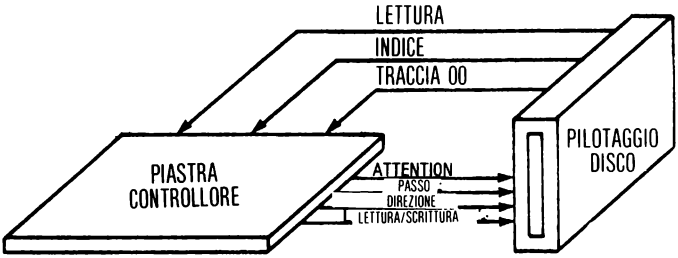


Fig. 4-90: Segnali di pilotaggio fondamentali del floppy

Segnali di stato del disco

La via PRONTO é vera quando il dischetto é stato correttamente inserito ed é pronto a spostarsi.

La via INDICE fornisce un impulso indicante l'inizio del settore 0. Un foro é realmente perforato nel dischetto, e rivelato da un circuito fotosensibile.

PROTEZIONE ALLA SCRITTURA (opzionale) indica al sistema che l'utente ha tolto via un tassello nel contenitore del dischetto per prevenire una scrittura accidentale.

Altre opzioni del disco

Alcune opzioni comuni sono:

PROTEZIONE ALLA SCRITTURA: Una linguetta speciale può essere rimossa dalla copertura del dischetto. Un sensore ottico nel circuito di pilotaggio la riconoscerà come fosse il tassello indicato. Essa protegge un dischetto da scritture accidentali (non disponibili in apparecchiature IBM).

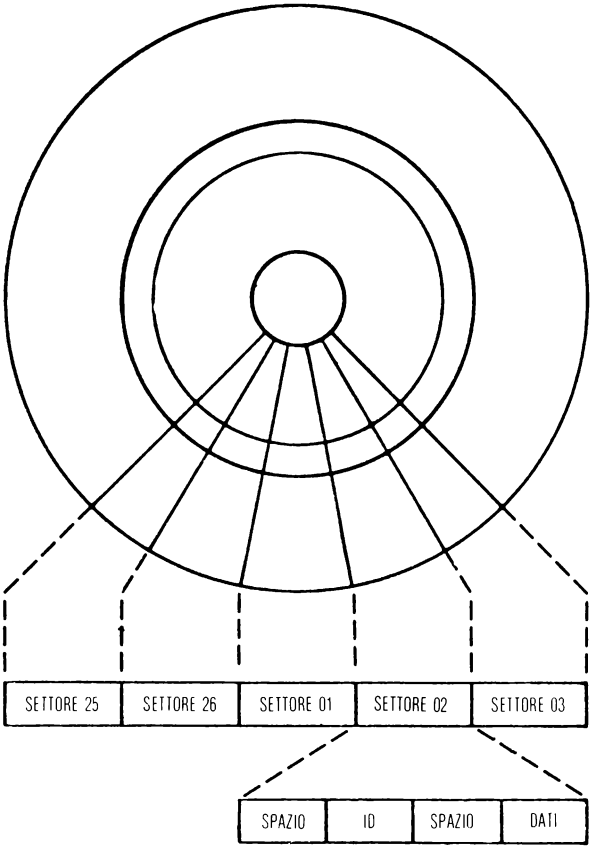


Fig. 4-91: Struttura di un settore

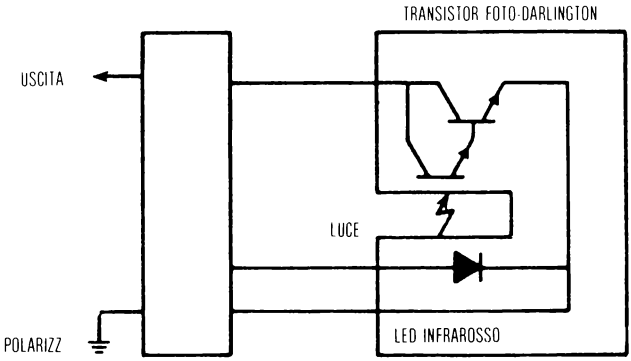


Fig. 4-92: Modulo di protezione alla scrittura

ESPULSIONE REMOTA: utile per assicurare che il dischetto è fuori dall'apparato di pilotaggio prima di spegnere il calcolatore, poichè i transistori di linea possono distruggere i contenuti.
STOP DEL MOTORE: riduce l'usura del motore, ma aumenta il tempo iniziale necessario per accedere al primo settore.
RICERCA AD ALTA VELOCITÀ: sposta la testina direttamente sulla traccia specificata, come ad esempio la 44esima. Essa richiede un registro differenza per la ricerca nell'apparato di pilotaggio.
SEPARATORE AD AGGANCIAMENTO DI FASE (o PLO): usualmente parte del controllore, piuttosto che del pilotaggio. Esso elimina il «jitter» dovuto agli sfasamenti dei picchi del segnale dati da quello di temporizzazione.

REGISTRAZIONE DELL'INFORMAZIONE

Ogni informazione è registrata in formato binario nelle tracce del disco. Normalmente è usata una tecnica NRZ (NRZ significa «Non Ritorno a Zero»): ciascuna direzione di bit è magnetizzata in una direzione («0»), o nell'altra («1»). Non esiste uno stato intermedio (uno «zero» significativo): da ciò NRZ in pratica è usata una codifica FM (modulazione di frequenza), che è autosincronizzante: ciascun bit del dato apparirà esattamente a metà di una «trama», cioè tra due successivi impulsi di temporizzazione.

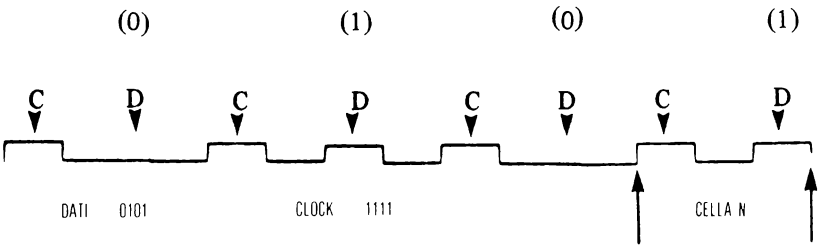


Fig. 4-93: Registrazione della informazione

In altre parole, ciascuna «trama» include un bit di temporizzazione (sempre un «1»), e un bit Dati («0» o «1»). Ciascuna trama dura quattro microsecondi e corrisponde ad una velocità di trasferimento di 250 kbps (la velocità di rotazione di un dischetto è standardizzata a 360 rpm ± 2%, prodotta da un segnale a 60 Hz).

Altri metodi di registrazione sono usati per aumentare la densità dei bit. Il principio base è di eliminare quanto possibile bit dati o di temporizzazione superflui. Tipicamente sono usati i metodi MFM = Modulazione di Frequenza Modificata o M2FM (MFM modificato) per dischetti a doppia densità.

La MFM è stata usata per pilotare dischi di alta efficienza come gli IBM 3330 e 3340.

Le regole della MFM sono:

1 - Il bit di dati appare ancora a metà del bit di trama

2 - Il bit di temporizzazione è scritto all'inizio della trama solo se si verificano due condizioni:

2-1 Non sono presenti bit dati nella trama corrente

2-2 Non erano presenti bit di dati nella trama precedente.

In altre parole il bit di temporizzazione è inserito soltanto se due trame consecutive contengono «00».

Nella lettura dei dati dal disco il segnale FM deve essere convertito in digitale, con assoluta precisione. Inoltre è necessaria una rivelazione separata per i bit di dati e di temporizzazione. Si possono presentare problemi speciali con alcune sequenze di bit. Ciò è conosciuto come il problema «scorrimento di bit», e per una precisa rivelazione del bit è usato il PLO (phase locked oscillator = oscillatore ad agganciamento di fase).

Tutti i dati del disco sono strutturati in bytes. I bytes (gruppi di 8 bit) devono essere anche sincronizzati. Questa funzione è ottenuta iniziando ciascun blocco di informazione con una speciale identificazione. Quando un dischetto è usato per la prima volta, esso deve essere inizializzato, o «formatted» con tali identificatori. I conteggi dei byte sono iniziati quando questi ID o Identificatori di Dati sono letti dal disco.

Infine, deve essere realizzata una conversione da serie a parallelo per assemblare gli 8 bit in un byte. Ciò è realizzato dal controllore del disco.

Le sequenze necessarie per una «scrittura» sono naturalmente quelle inverse a quelle descritte per la «lettura».

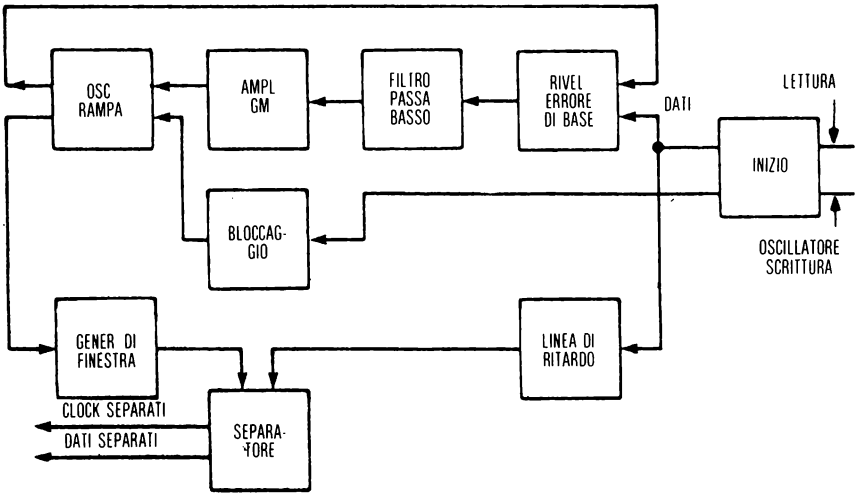


Fig. 4-94: Un PLO discreto

L'oscillatore ad agganciamento di fase (PLO)

Un circuito reazionato ad agganciamento di fase (PLO) assolve la funzione di sincronizzare i segnali di temporizzazione della lettura e della scrittura. I dati in uscita dal PLL contengono CLOCK + DATI decodificabili dal formato NRZ del disco. L'uscita di temporizzazione del PLL genera il necessario fronte del clock al centro della cella del bit.

Programma base del FDC

Il programma base segue una sequenza di quattro passi:
1 - Inizializzazione del FDC.
2 - Ricerca della traccia 0. Controllo di errori.
3 - Scrittura di un settore in una traccia. Controllo di errori.
4 - Lettura dello stesso settore. Controllo di errori.

Filosofie per il caricamento della testina

Due filosofie fondamentali sono usate:
1 - Caricamento continuo. Ciò comporta una usura continua.
2 - Caricamento per il tempo minimo. Ciò comporta frequenti caricamenti e disimpegni.

Accensione

Alla accensione il programma di inizializzazione porta la testina sulla traccia 00 (normalmente la traccia indice, alla quale occorre accedere per prima). Il contatore di settore, dopo l'accensione, contiene il numero del settore a meno di una rotazione.

Aggiornamento di un settore

Una volta che al dischetto è stata data la struttura dei formati gli unici campi alterabili sono lo spazio di identificazione, il campo dati e il primo byte, o spazio dati.

Settorizzazione hardware e settorizzazione software

Un dischetto con settorizzazione hardware ha fori per 32 settori che definiscono settore di 32 x 128 bytes. Eliminando la necessità di controllare dei settori è possibile memorizzare piú informazione.

Un dischetto con settori in software è IBM compatibile, con settori di 26 x 128 bytes. I settori devono essere identificati da una testata. Possono essere memorizzati meno dati, ma è accresciuta l'affidabilità e la flessibilità.

Densità doppia, doppia faccia, pilotaggio doppio: caratteristiche

Sono usate due tecniche per accrescere l'ammontare di informazione che può essere memorizzata nel floppy: doppia densità e doppia testina. La densità doppia

raddoppia il numero di bit per traccia usando una tecnica di registrazione «compatta», quella MFM (M2FM).

Essa richiede tolleranze molto ridotte per un funzionamento affidabile, ed è molto meno tollerante verso variazioni di velocità rispetto ad una regolare codifica FM.

Sono necessarie testine doppie per utilizzare entrambe le facciate del floppy. Le testine sono posizionate con uno sfasamento reciproco di 180°. Ciò accresce la complessità meccanica e il costo del pilotaggio, oltre all'usura.

Entrambe le tecniche sono comunemente usate per raddoppiare il numero di bytes per dischetto.

Nella tecnica del pilotaggio doppio si fa uso di due dischetti, ma di un singolo motore dell'albero e di un singolo apparato di posizionamento della testina. Esso è molto piú economico di due distinti circuiti di pilotaggio, e in qualche modo piú lento.

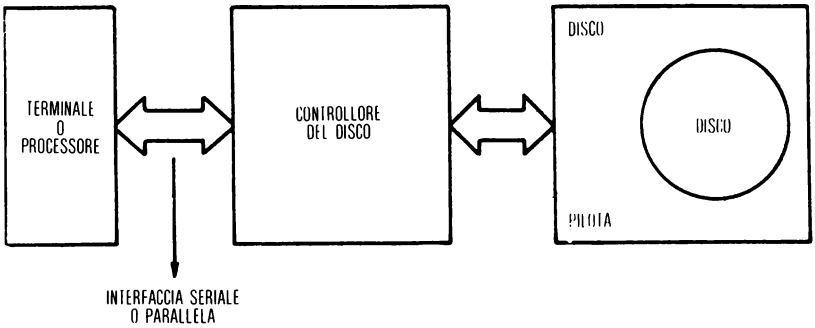


Fig. 4-95: Interfacce del controllore di disco tra il pilotaggio e il processore

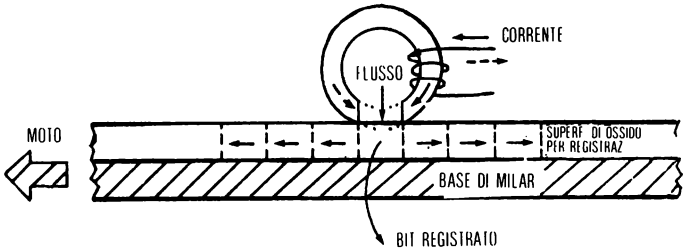


Fig. 4-96: Memorizzazione di un bit sul disco

Formati del disco

Sia l'informazione di temporizzazione che i dati sono codificati nello stesso segnale. Gli impulsi di temporizzazione sono presenti in ogni bit. Un dato «0» è indicato dall'assenza di un altro impulso durante l'intervallo di una cella di un bit. Ciò è indicato in figura 4-97. Un «1» è indicato da un impulso dati a metà dell'intervallo della cella del bit.

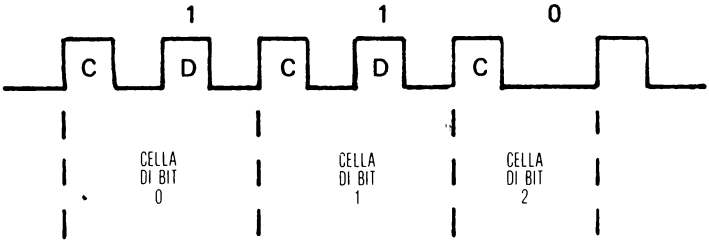


Fig. 4-97: Rappresentazione del clock e dei dati

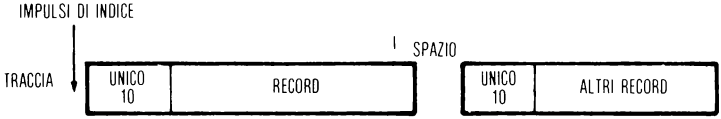


Fig. 4-98: Identificatore di record

La *settorizzazione in software* significa che la suddivisione del disco o della traccia è realizzata in *software*. Ciò è in contrapposizioine alla settorizzazione hardware, nella quale l'inizio di ciascun settore è fisicamente delineato mediante un foro perforato nel disco. Nella settorizzazione software, ciascuna traccia ha inizio con un *impulso-indice* fisico, corrispondente alla rivelazione del foro indice nel disco. Ciascun record è preceduto da un unico *identificatore*. Vedi figura 4-98. Record successivi sono separati da spazi (gaps). Tali spazi sono necessari per aggiornare l'informazione senza cancellare il record precedente o seguente. A causa delle ridotte variazioni di velocità nel motore di pilotaggio del disco, ogni volta che i contenuti totali o parziali del disco devono essere modificati, e quindi riscritti, la fine

	DATI	CLOCK
Segno indice di indirizzo	FC	D7
Segno ID di indirizzo	FE	C7
Stato di indirizzo di dati	FB	C7
Segno di indirizzo di dati cancellati	F8	C7

Fig. 4-99: Indicatori di indirizzo

del record può estendersi oltre la fine di quello precedentemente registrato.

Per questa ragione deve essere presente uno spazio vuoto tra la fine di un record e l'inizio del successivo. Infatti deve essere presente una spaziatura tra due zone che possono essere modificate separatamente. Molto spesso è usato il formato della traccia di disco IBM, a volte con piccole variazioni. Questo formato è illustrato in figura 4-100. Quattro tipi di spaziature sono usate:

Lo spazio 4 è usato una sola volta nella traccia. Esso è lo spazio di indice libero. Esso è presente alla fine della traccia, appena prima della posizione foro-indice.

Lo spazio 1 è chiamato lo spazio-indice ed è usato all'inizio di ciascuna traccia. Esso contiene 20 bytes: i primi 16 bytes contengono la sequenza esadecimale «FF» seguita da 4 bytes che contengono «00». Questi quattro bytes «0» costituiscono il metodo classico per fornire la sincronizzazione per il separatore dei campi dati. La lunghezza dello spazio 1 è sempre fissa. Il foro indice è seguito dalla identificazione del primo record.

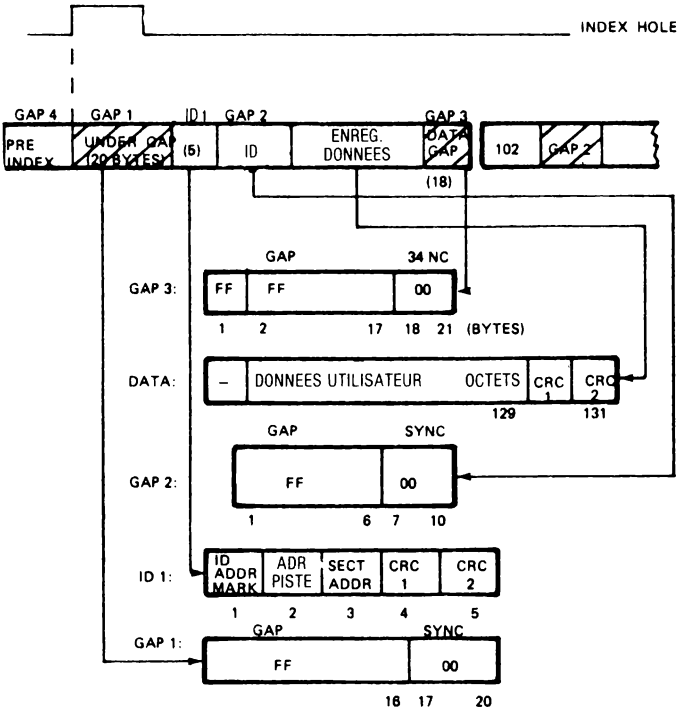


Fig. 4-100: Formato di disco floppy IBM

IDI è il campo di identificazione del primo record. Esso è composto da 5 bytes: segno di indirizzo ID, indirizzo di traccia, indirizzo di settore, e due bytes CRC di tipo controllo somma per verificare l'integrità del campo. L'indirizzo di traccia e quello di settore eseguono una verifica che l'accesso sia stato eseguito sulla traccia e sul settore corretti.

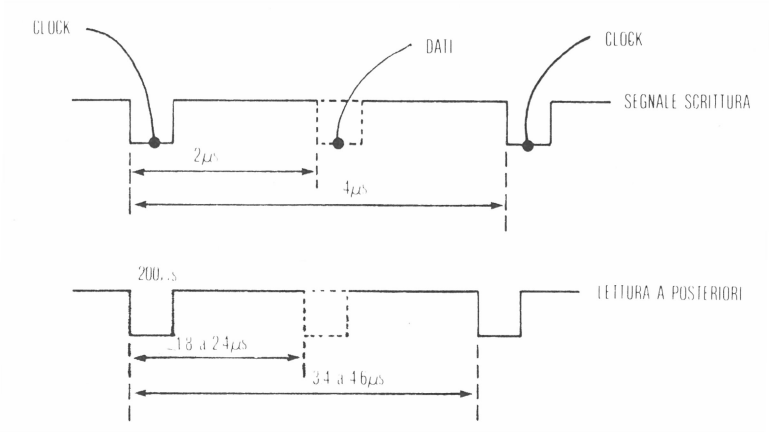


Fig. 4-101: Temporizzazione

È chiamato *Gap 2* lo spazio ID che separa ciascun campo di identificazione dal suo campo dati. Esso è composto da 10 bytes. I primi 6 bytes contengono la sequenza esadecimale «F». Essa è seguita dai 4 bytes «00», usati come sempre per la sincronizzazione. La lunghezza del Gap 2 può variare la sua lunghezza dopo aggiornamenti.

Segue il primo *record*, o campo dati. Esso è composto da 131 bytes (vedi figura 4-100). I primi bytes contengono dati o indicatori di indirizzo cancellati. Seguono i 128 bytes dei dati di utente. Seguono nel record i due bytes CRC di controllo della somma.

Infine, il *Gap 3*, termina il primo record. Esso è chiamato lo spazio-dati e si compone di 18 bytes. I primi 17 bytes sono la sequenza «FF», mentre i 4 bytes successivi, «00», sono di sincronizzazione. Ciascun record successivo, o settore, inizia con ID, spazio 2, e cosí via.

Settorizzazione hardware

Quando è usata la settorizzazione hardware, è usato un dischetto e un pilotaggio speciali. Sul disco è perforato un foro all'inizio di ciascun settore. Ciascuno di essi ha pertanto inizio da un impulso di settore generato fisicamente. Nel caso del mini-

disco floppy sono usate due configurazioni: 16 settori di 128 bytes o 10 settori di 256 bytes per traccia. La traccia ha inizio con un impulso indice.
Ciò è illustrato in figura 4-102.

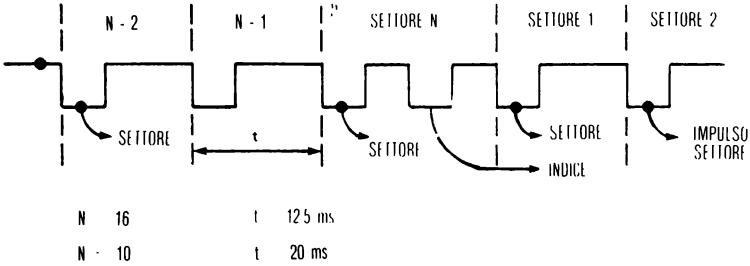


Fig. 4-102: Temporizzazione del disco con settori generati in hardware

RIVELAZIONE E CORREZIONE DI ERRORE

Tre tipi di errore sono individuati.

Errore di scrittura

Esso corrisponde al caso nel quale i dati non sono stati scritti correttamente nel disco. Il modo per verificare se i dati sono stati scritti correttamente consiste nell'uso della procedura «controllo scrittura», mediante la quale i dati sono letti durante la successiva rotazione del disco. Normalmente, l'utente riscriverà i dati che non sono stati scritti correttamente nel disco, ripetendo la procedura fino a 10 volte. Se non si riesce a scrivere il settore o la traccia saranno considerati non usabili.

Errore di lettura

Essi sono distinti in due tipi:

1. Software: ciò corrisponde al caso in cui l'errore é transitorio ed é corretto semplicemente mediante una seconda lettura (e ancora fino a 10 volte) oppure riposizionando la testina.

Più comunemente è mossa la testina un passo avanti nella direzione precedente e poi riportata indietro.
In tal modo si correggono gran parte degli errori. Se tale procedura non ha esito positivo, l'errore è di tipo hardware:

2. Hardware: quando le usuali procedure di correzione non hanno buon esito nella lettura dei dati dal disco, esso é da considerare inutilizzabile. I dati sono perduti.

Errore nella ricerca

Esso corrisponde al caso in cui la testina non si posiziona sulla traccia corretta. Ciò può essere verificato mediante la lettura del campo ID all'inizio della traccia,

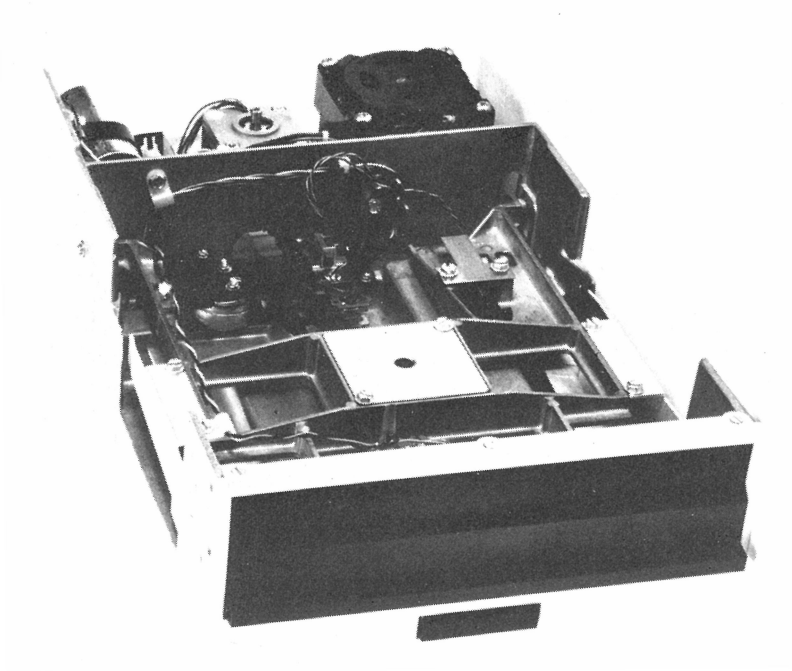


Fig. 4-103: Figura di una unità di pilotaggio di un floppy a doppia faccia

che contiene l'indirizzo di traccia. Quando é rivelato un errore il contatore di traccia del pilotaggio del disco deve essere ricalibrato. La testina é riportata indietro sulla traccia 00 ed é dato un'altro ordine di ricerca.

RIVELAZIONE DEGLI ERRORI

La tecnica sempre usata per la rivelazione dei dati scritti in modo errato nel disco é quella del tipo a *controllo della somma*. A tale scopo é usato il controllo a ridondanza ciclica (CRC = Cyclic-Redundancy-check). Ciascun campo termina con due byte CRC. I bit dei dati sono divisi per il polinomio generatore $G(X) = X^{16} + X^{12} + X^3 + 1$.

Il resto di tale divisione é chiamato CRC. Esso é aggiunto nei due bytes che seguono i dati. Nella lettura dei dati dal dischetto é letto anche il CRC. Esso é diviso insieme ai dati per il polinomio G(X) e, se il resto della divisione é diverso da zero, é indicata la presenza di un errore.

Esistono circuiti CRC implementati in un singolo circuito integrato, come il FAIRCHILD 9401, il Motorola 8501, e altri, che rivelano tali errori mediante un singolo integrato. Esistono anche controllori di disco floppy che comprendono la funzione CRC nel singolo circuito integrato nel quale sono realizzati.

Controllo mediante ridondanza ciclica

Il CRC é il metodo favorito per verificare l'integrità di aree di memoria con un numero minimo di bit errati. La parità rivela un errore su singolo bit entro una parola. Quando essa non é disponibile, o é implementabile in modo costoso, é usato il CRC per rivelare errori in un blocco di parole. Il CRC é in particolare quasi universalmente usato nel caso del disco floppy, e nelle cassette a nastro magnetico. Esso é anche usato per verificare l'integrità delle ROM.

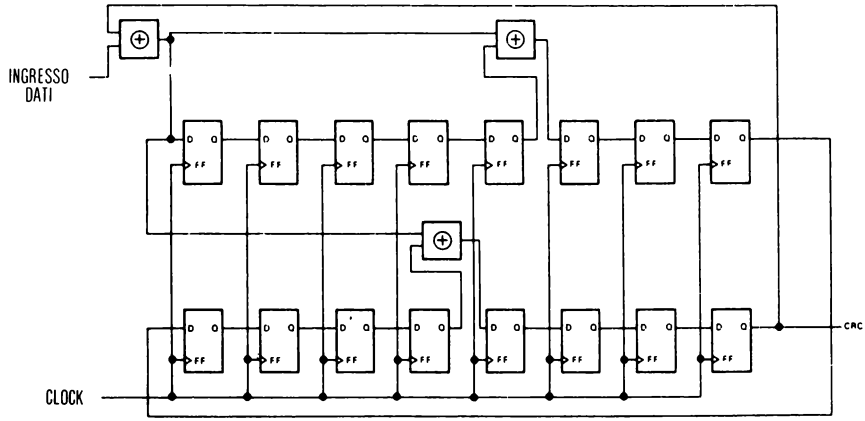
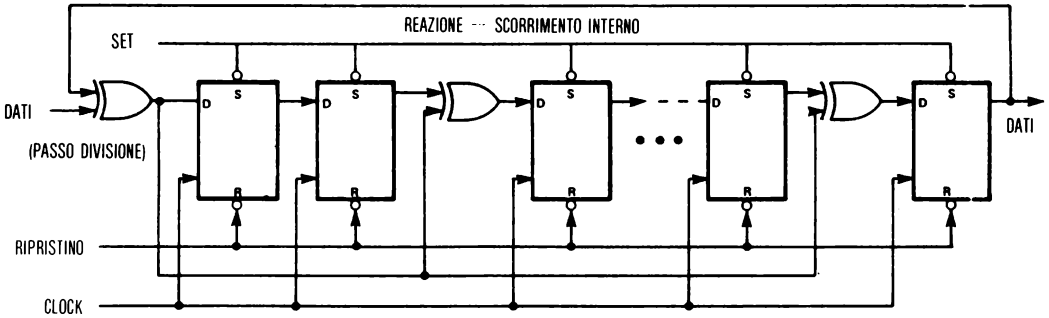


Fig. 4-104: Dettaglio dell'hardware di verifica con CRC



HARDWARE DI GENERAZIONE DA CRC

$(x^{16} + x^{15} + x^{2} + 1)$

Fig. 4-105: Dettaglio dell'hardware di generazione del CRC

Il principio della tecnica CRC é la seguente: gli 8 bit della parola sono trattati come coefficienti di un polinomio di grado 7.

La sequenza di bit $B_7\ B_6\ B_5\ B_4\ B_3\ B_2\ B_1\ B_0$ é interpretata come $B_7 X^7 + B_6 X^6 + B_5 X^5 + B_4 X^4 + B_3 X^3 + B_2 X^2 + B_1 X^1 + B_0 X^0$. X é qui chiamata una variabile fantoccio.

Per esempio, la parola binaria 10000011 rappresenta:

$B(X) = 1 X^7 + 0 X^6 + 0 X^5 + 0 X^4 + 0 X^3 + 0 X^2 + 1 X^1 + 1 X^0 = X^7 + X^2 + 1$

Sarà usato un generatore del polinomio G(X). Il polinomio B(X) corrispondente alla parola binaria é diviso per il polinomio generatore G(X). Il risultato é il quoziente Q(X) e il resto R(X). $B(X) = G(X)\ Q(X) + R(X)$.

Il risultato del controllo mediante ridondanza ciclica é di porre in coda ad una stringa di bit uno (o più) bytes aggiuntivi, cioé R(X), in modo che la stringa totale sia esattamente divisibile per il polinomio generatore. La precedente equazione può essere riscritta: $B(X) - R(X) = Q(X) \cdot G(X)$. La stringa composta da B e dal resto R è esattamente divisibile per G (X). I bit extra aggiunti alla stringa B sono chiamati bit (o bytes) CRC. Ricevendo inizialmente una stringa B, il generatore CRC calcola il resto R che sarà aggiunto alla stringa.

Quando la stringa é richiamata dal disco, é letta la sequenza completa di bits inclusi quelli del CRC. Essi sono perfettamente divisibili per il polinomio generatore G(X). Se essi non lo sono, é indicata la presenza di un errore. Se il resto della divisione é zero, o non si é verificato errore o esso non é riconoscibile.

Come al solito, l'algoritmo CRC può essere realizzato sia in hardware che in software. Sono disponibili generatori CRC in un singolo circuito integrato. Un esempio di CRC hardware é indicato in figura 4-105. Una implementazione software del CRC, che usa un Signetics 2650, é indicata in figura 4-106. La divisione in hardware é in quel caso realizzata mediante un registro a scorrimento con reazione. Il generatore CRC corrispondente alla illustrazione é $G(X) = X^{16} + X^{15} + X^2 + 1$. La retroazione in OR-esclusivo realizza la divisione durante gli scorrimenti successivi attraverso i flip-flop del registro.

CONCLUSIONI SUL FUNZIONAMENTO DEL DISCO

Tutti i principi di funzionamento del disco floppy sono stati spiegati. In particolare sono stati indicati i segnali per pilotare il disco, l'intervento risultante, il formato dei dati, e il meccanismo di controllo di errore. Sarà ora descritta l'implementazione di un apparato di controllo per pilotaggio del disco da interfacciare con un sistema a microprocessore.

SEZIONE II: CONTROLLORE SHUGART SA 4400

Questa piastra di controllo é realizzata con il circuito integrato controllore bipolare 300 SMS della Signetics. Esso é progettato per controllare 1, 2, o tre mini-floppy SA400. Esso sarà qui brevemente descritto, per mostrare la capacità di un intero controllore di mini-floppy. Pertanto saranno presentati altri progetti compatti

```
* CYCLIC REDUNDANCY CHECK SUBROUTINE (SIGNETICS 2650)
*
* THIS ROUTINE GENERATES A 16-BIT CHECK CHARACTER FOR
*   THE DATA CHARACTER IN RØ; VARIOUS POLYNOMIALS
*   CAN BE ACCOMODATED BY CHANGING THE CONSTANTS
*   SPECIFIED AT PROGRAM LOCATIONS CKØ AND CK1 AS PRR
*   THE TABLE BELOW
*
* DEFINITION OF SYMBOLS
*
RØ      EQU     Ø       PROCESSOR REGISTERS
R1      EQU     1
R2      EQU     2
WC      EQU     H'Ø8'   PSL:  1=WITH,Ø=WITHOUT CARRY
C       EQU     H'Ø1'         CARRY/BORROW
UN      EQU     3       BRANCH CONDITION  UNCONDITIONAL
EQ      EQU     Ø                         EQUAL
*
* TABLE OF POLYNOMIALS
*
CRCFØ   EQU     H'4Ø'   CRC16 FORWARD
CRCF1   EQU     H'Ø2'
CRCRØ   EQU     H'2Ø'   CRC16 REVERSE
CRCR1   EQU     H'Ø1'
CCIFØ   EQU     H'Ø8'   CCITT FORWARD
CCIF1   EQU     H'1Ø'
CCIRØ   EQU     H'Ø4'   CCITT REVERSE
CCIR1   EQU     H'Ø8'
*
* BEGINNING OF SUBROUTINE
*
        ORG Ø
*                       INITIALIZATION
CRCGEN  PPSL    WC      OPERATIONS WITH CARRY
        LODI,R2 8       INITIALIZE BIT COUNTER
        LODA,R1 CRC+1   GET OLD REMAINDER LSB
        EORA,RØ CRC     EX-OR OLD REMAINDER MSB WITH DATA
*
TEST    CPSL    C       CLEAR CARRY
        TMI,RØ  H'8Ø'   TEST MS-BIT OF RØ
        BCFR,EQ SHIFT   BRANCH IF NOT A '1'
        PPSL    C       PRESET CARRY
CKØ     EORI,RØ CRCFØ   APPLY 'FEEDBACK'
CK1     EORI,R1 CRCF1
*
SHIFT   RRL,R1          SHIFT THE DOUBLE CHARACTER
        RRL,RØ
        BDRR,R2 TEST    CHECK IF DONE
        STRA,RØ CRC     SAVE THE NEW REMAINDER
        STRA,R1 CRC+1
        RETC,UN
*
* RAM AREA
*
        ORG     H'5ØØ'
CRC     RES     2       REMAINDER MSB IN CRC
        END     CRCGEN
```

Fig. 4-106: Programma di CRC del 2650

che fanno uso del nuovo circuito FDC.

Il controllore é compatibile con il formato IBM 3740, ma usa una struttura degli spazi modificata (lo spazio pre-indice, lo spazio 4, é più corto). Esso é provvisto di una memoria temporanea per i dati di 128 bytes. Sono fornite 8 funzioni di controllo:

- INIT: esso ripristina il controllore nel disco.
- SEK: sposta la testina sulla traccia specificata.
- READ legge un settore (128 bytes).
- READ ID: legge il successivo identificatore di settore.
- WRITE: scrive dati in un settore (128 bytes) con i dati di indicazione di indirizzo (AM).

Gli ultimi tre comandi leggono o scrivono dati tra il processore e la memoria temporanea del disco, o tra la memoria e il disco.

- WRITE-DDL: svolge la stessa funzione del comando WRITE ma dopo aver cancellato l'AM (address mark).
- FORMAT: scrive l'indicatore dati AM, gli spazi, i dati nell'intera traccia con il formato 3740.
- STATUS: indica lo stato al circuito di pilotaggio.

I segnali usati dalla interfaccia 4400 per comunicare con il sistema microprocessore sono illustrati in figura 4-107. La sequenza fondamentale degli eventi realizzata dal controllore è semplicemente:

1. Ricercare la traccia.
2. Trovare il settore.
3. Far scorrere e trasferire il contenuto del numero desiderato di settori.
4. Controllare il CRC.

Alcuni comandi sono necessari per il funzionamento del controllore. Gran parte dei controllori forniscono soltanto da sei a dieci comandi.

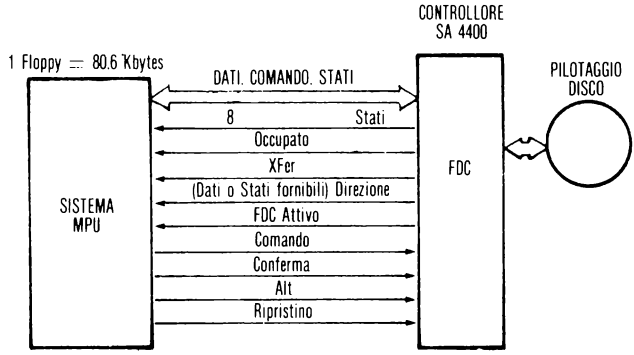


Fig. 4-107: Segnali di interfaccia

La piastra SA 4400 FDC SHUGART

Essa può contenere da uno a tre circuiti di pilotaggio con posizionamento di testina concomitanti. Fornisce 8 comandi.

È usata logica bipolare. Gli otto comandi sono:

INIT	– reset di sistema
SEEK	– posizionamento della testina sulla traccia
READ	– lettura di settore di disco
READID	– lettura del nuovo ID
WRITE	– scrittura di settore di disco
WRDEL	– scrittura di settore relativo a dati cancellati
FORMAT	– scrittura di indicatori di indirizzo, spazi, e dati sulla intera traccia
STATUS	– risposta di indicazione di stato verso il circuito di pilotaggio indirizzato.

SEZIONE III: INTEGRATO WD 1771 CONTROLLORE DI DISCO FLOPPY

Questo controllore-realizzatore di formato per disco floppy, realizzato in un singolo integrato, interfaccia gran parte dei costruttori di unità di pilotaggio ed è naturalmente compatibile con l'IBM 3740.

Esso fornisce:

- ricerca automatica di traccia con verifica. Questa caratteristica deve essere presente in tutti i FDC.
- compatibilità di formato per settori definiti in software. Questa caratteristica potrebbe essere standard nei FDC.
- lettura o scrittura con:
  - record singolo o multiplo
  - ricerca automatica di settore
  - lettura o scrittura di una intera traccia.

Tutte le caratteristiche indicate potrebbero essere standard nei FDC.

- controlli programmabili:
  - tempo di passo da traccia a traccia
  - tempo di posizionamento di testina
  - tempo di impegno testina
  - controllo del motore a tre fasi oppure con passo piú direzione
  - trasferimento DMA o mediante programma

Il lettore esperto osserverà che tutte le caratteristiche indicate sono essenzialmente standard in tutti i FDC. Le differenze consistono normalmente nel livello e il numero di moduli di pilotaggio di disco che un singolo integrato è capace di controllare simultaneamente.

L'architettura interna dello FD1771B è indicata nella figura 4-108, e sarà ora descritta in dettaglio. Esso contiene essenzialmente 5 circuiti funzionali, sei registri,

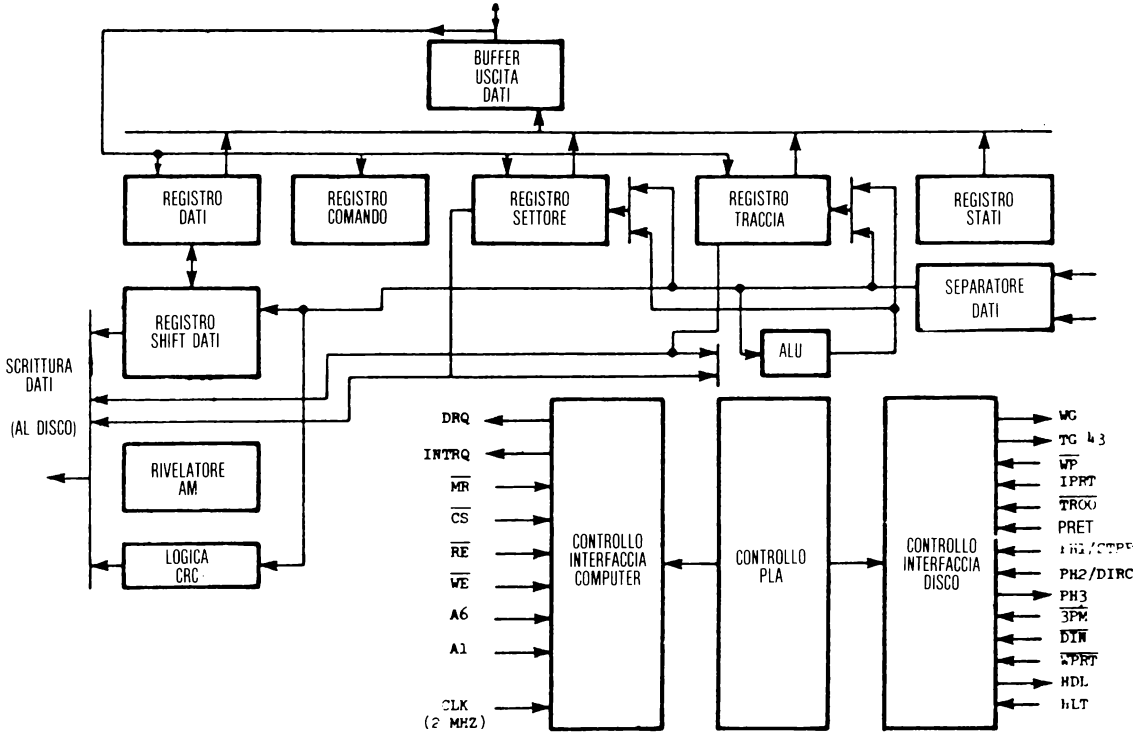


Fig. 4-108: Integrato RDC della Western Digital

e due interfacce: una interfaccia per il processore e una interfaccia per il disco floppy. Ciascuno di essi è di seguito esaminato.

I quattro circuiti funzionali

I quattro circuiti essenziali, che appaiono nell'illustrazione, sono:

- la logica CRC che genera il carattere di controllo
- la ALU (Unità Logica Aritmetica/arithmetic-logic-unit) che è stata usata per le comuni funzioni aritmetiche. In particolare, per confrontare caratteri relativi a contenuti di spostamento.
- il controllo dell'interfaccia del disco
- il controllo della interfaccia del calcolatore.

Entrambe le interfacce sono di seguito descritte.

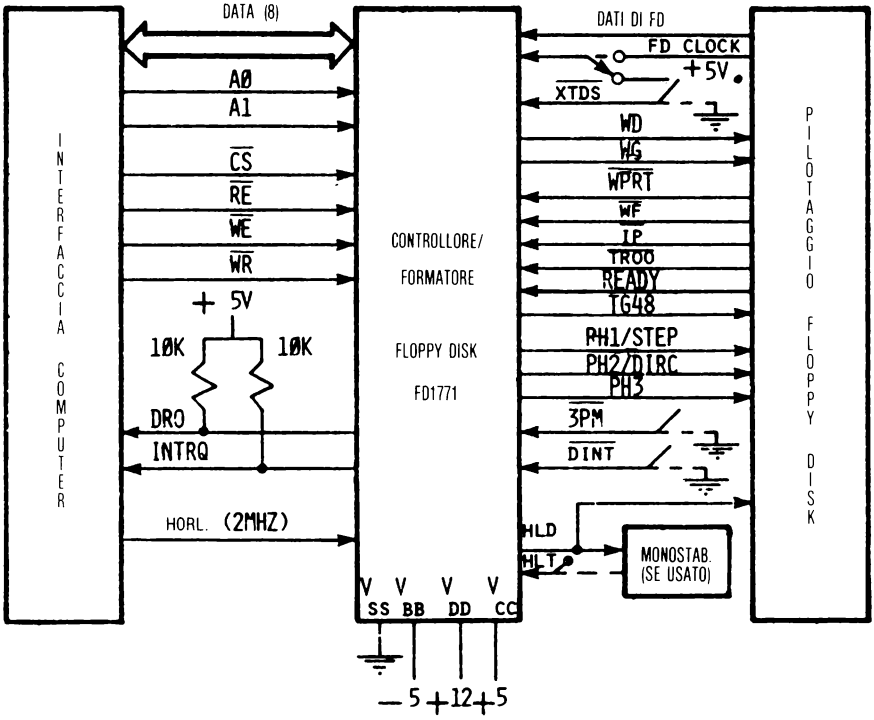


Fig. 4-109: Interfaccia del disco floppy mediante l'FD1771

I sei registri interni

Da sinistra a destra nella figura 4-108 è possibile distinguere:

1. *registro scorrimento dati:* raccoglie 8 bit dai dati del disco floppy, o mette in serie 8 bit ricevuti dal bus-dati del microprocessore sulla via dei dati del disco floppy.
2. *registro dati:* è soltanto un registro per contenere un byte durante l'operazione di scrittura e di lettura. Comunica con la memoria temporanea per i dati in uscita e può ricevere i dati direttamente dal bus dei dati del microprocessore.
3. *registro comandi:* è usato per contenere gli 8 bit di comando che devono essere eseguiti. Questo registro è caricato dal programmatore e specifica il modo di operare del disco.
4. *registro di settore:* contiene l'indirizzo della posizione di settore desiderata.
5. *registro di traccia:* contiene il numero di traccia della relativa posizione corrente. Un incremento corrisponde ad uno spostamento verso l'interno (fino a 76 tracce in un disco di dimensione regolare), un decremento all'opposto.
6. *registro di stato:* mantiene l'informazione di stato del controllore.

Interfaccia del processore

L'interfaccia del processore e quella del disco floppy sono illustrate in figura 4-109. Il FDC comunica con il processore attraverso 8 vie dei dati bidirezionali chiamati DAL (Data Access Line = via di accesso dati). È specificato un ingresso quando $\overline{CS}$ e $\overline{WE}$ (write enable = abilitazione scrittura) sono attive. Una uscita è specificata quando $\overline{CS}$ e $\overline{RE}$ (read enable = abilitazione lettura) sono attive. La destinazione interna è specificata da A1 - A0 in accordo alla tabella seguente. L'uscita richiesta dati (DRO) è usata per il DMA. La richiesta di interruzione (INTRT) è attivata da varie condizioni.

(PAROLA COMANDO) BIT 1	BIT 0	PERIODO (MS)	VELOCITÀ (PASSI AL SECONDO)
0	0	6	166
0	1	6	166
1	0	8	125
1	1	10	100

Fig. 4-110: Bit della parola di comando

Interfaccia del disco floppy

I segnali sono indicati sulla destra della figura 4-109. Essi forniscono il controllo del posizionamento della testina, controlli di scrittura e trasferimenti dati. Il segnale di temporizzazione è a onda quadra a 2 MHz, internamente diviso per 4, producen-

do 500 kHz. Esso permette tre velocità di passo programmabili, controllate dal bit 0 e dal bit 1 della parola di comando in accordo alla tabella seguente.
Il tempo di assestamento della testina è addizionale e vale 10 millisecondi.

A1	A0	$\overline{RE}$	$\overline{WE}$
0	0	Registro di stato	Registro comandi
0	1	Registro di traccia	Registro di traccia
1	0	Registro di settore	Registro di settore
1	1	Registro dati	Registro dati

Fig 4-111: Indirizzamento di registro

Funzionamento del disco

Un'*operazione di lettura* dal disco è realizzata in 5 fasi:

1. Caricamento del registro di traccia
2. Comando di ricerca
3. Attesa di verifica
4. Trasferimento dei dati al microprocessore mediante controllo di interruzione
5. Verifica dell'interruzione dopo il numero corretto di trasferimenti

In corrispondenza, una *operazione di scrittura* è realizzata in sette fasi:

1. Caricamento del registro di traccia
2. Comando di ricerca
3. Attesa di verifica
4. Comando di scrittura
5. Caricamento del primo dato dopo che la richiesta dati è stata ricevuta
6. Caricamento dei dati rimanenti
7. Verifica del semaforo $\overline{BUSY}$ e errore CRC

Conclusioni

Il modulo FD1771D indica come è possibile integrare gran parte delle funzioni richieste per il controllo mediante un singolo integrato di un disco floppy normale. Esso fornisce essenzialmente tutte le caratteristiche necessarie per controllare e realizzare il formato del disco.

SEZIONE IV: INTEGRATO WD 1781 CONTROLLORE DI DISCO FLOPPY

Questo FDC in unico integrato è la versione doppia del 1771, del quale è secondo fornitore la National Semiconductor. Esso fornisce entrambi i formati a densità semplice e doppia. Nel modo a densità doppia, la codifica/decodifica deve essere

realizzata dai circuiti dell'utente per il ripristino dei dati. In tal modo il modulo 1781 è utilizzabile con entrambe la MFM e la M2FM.

Lista dei comandi dello FD 1781

1. Ripristino
2. Ricerca
3. Passo
4. Passo interno
5. Passo esterno
6. Comando lettura
7. Comando scrittura
8. Indirizzo lettura
9. Traccia di lettura
10. Traccia di scrittura
11. Interruzione forzata

Errori hardware

Gli errori tipici del disco possono essere classificati nel modo seguente:

1. READ (lettura)

   11. Dati non accessibili

   Può essere impossibile recuperare i dati dal disco, a causa di una cattiva registrazione, rumore, difetti di superficie, o sporcizia. Ciò è rivelato da caratteri di controllo somma o CRC. Il controllore/realizzatore di formato calcola il proprio controllo di somma o CRC durante una operazione di lettura. Esso quindi confronta il CRC risultante dalla lettura dei dati del disco con quello calcolato. Una discrepanza indica la presenza di un errore.

   12. settore o traccia errati

   Esso è un errore di posizionamento della testina. Esso può essere dovuto ad una errata frequenza dell'impulso di passo, o rumore nel contatore. Nel caso di un'errore di traccia, la traccia sarà riposizionata alla traccia 00 (equipaggiata di uno speciale sensore), e riposizionata nuovamente. Naturalmente, la rivelazione di un tale errore implica che il settore e la traccia di 10 sono memorizzati all'inizio di ciascun settore. Questa informazione è sempre letta e controllata dal controllore, prima di usare il settore.

2. WRITE

   L'errore è indicato da un semaforo FILE UNSAFE («file» insicuro). Esso si presenta quando è fatto un tentativo di scrittura in un dischetto protetto contro la scrittura, con una porta di pilotaggio aperta, durante un malfunzionamento della elettronica di pilotaggio.

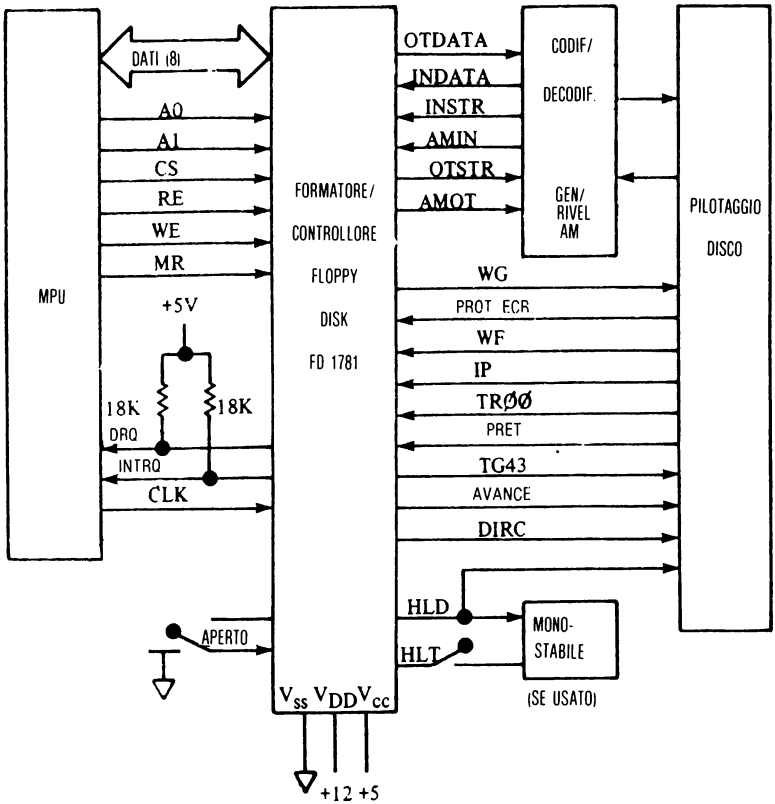


Fig. 4-112: Diagramma a blocchi del sistema WD 1781

PIASTRA DEL CONTROLLORE PERSCI

Il controllore PerSci usa il modulo WD1771 e un 8080 per realizzare le funzioni di controllo intelligente di disco floppy. La piastra contiene il separatore dati PLO, il modulo di interfaccia 1771, una memoria temporanea RAM statica di 1 K, una memoria ROM di 4K, la CPU 8080, e circuiteria di vario tipo.

Fisicamente, il circuito consiste di una piccola piastra di circuito stampante. Per adattare il controllore al bus S100 può essere realizzato un particolare artifizio ponendo il controllore sulla piastra standard del bus S100. I dati sono registrati nel formato standard IBM e possono essere adatti fino a 4 unità di pilotaggio.

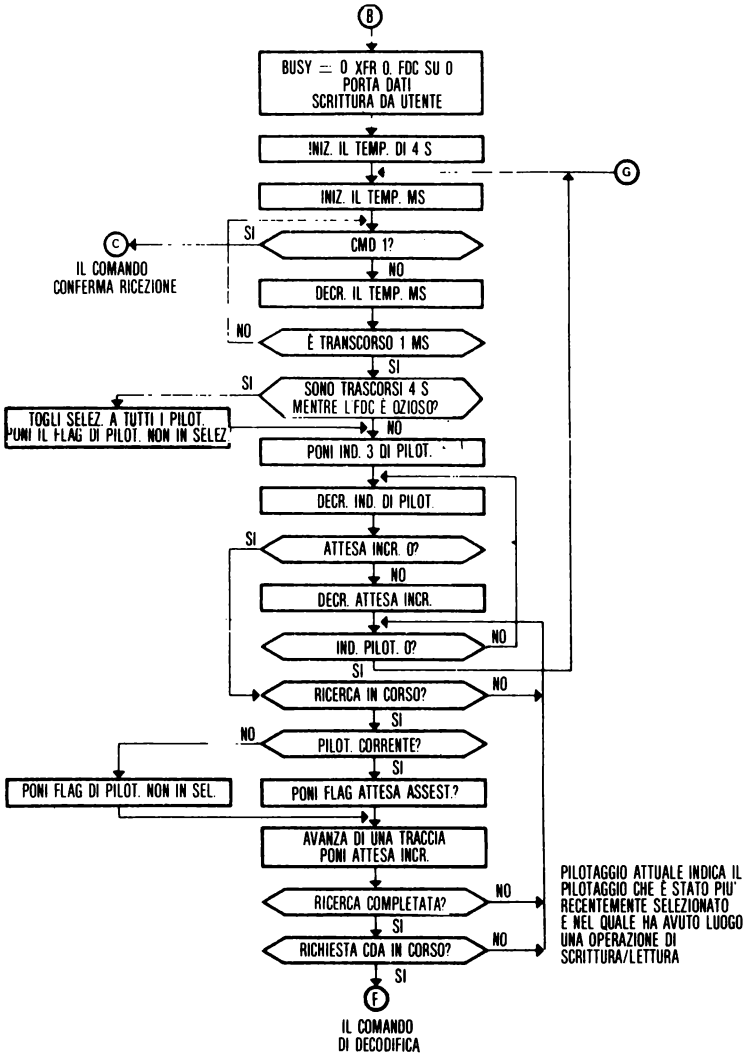


Fig. 4-113: Diagramma di flusso del software della PerSci

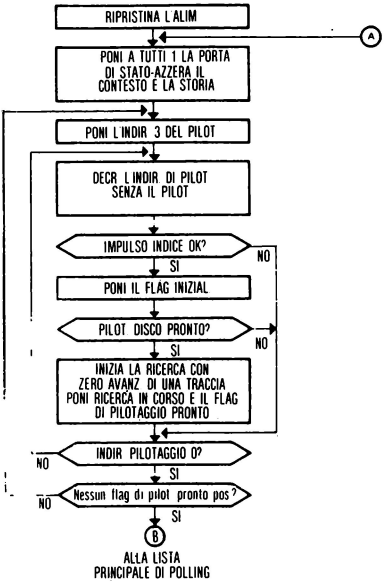


Fig. 4-114: Continuazione del diagramma di flusso

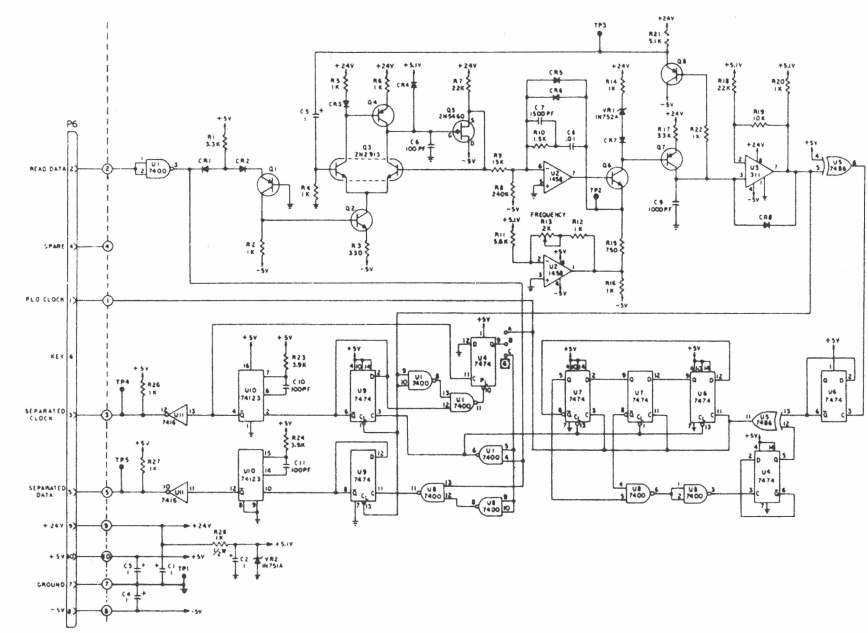


Fig. 4-115: PLO separatore di dati

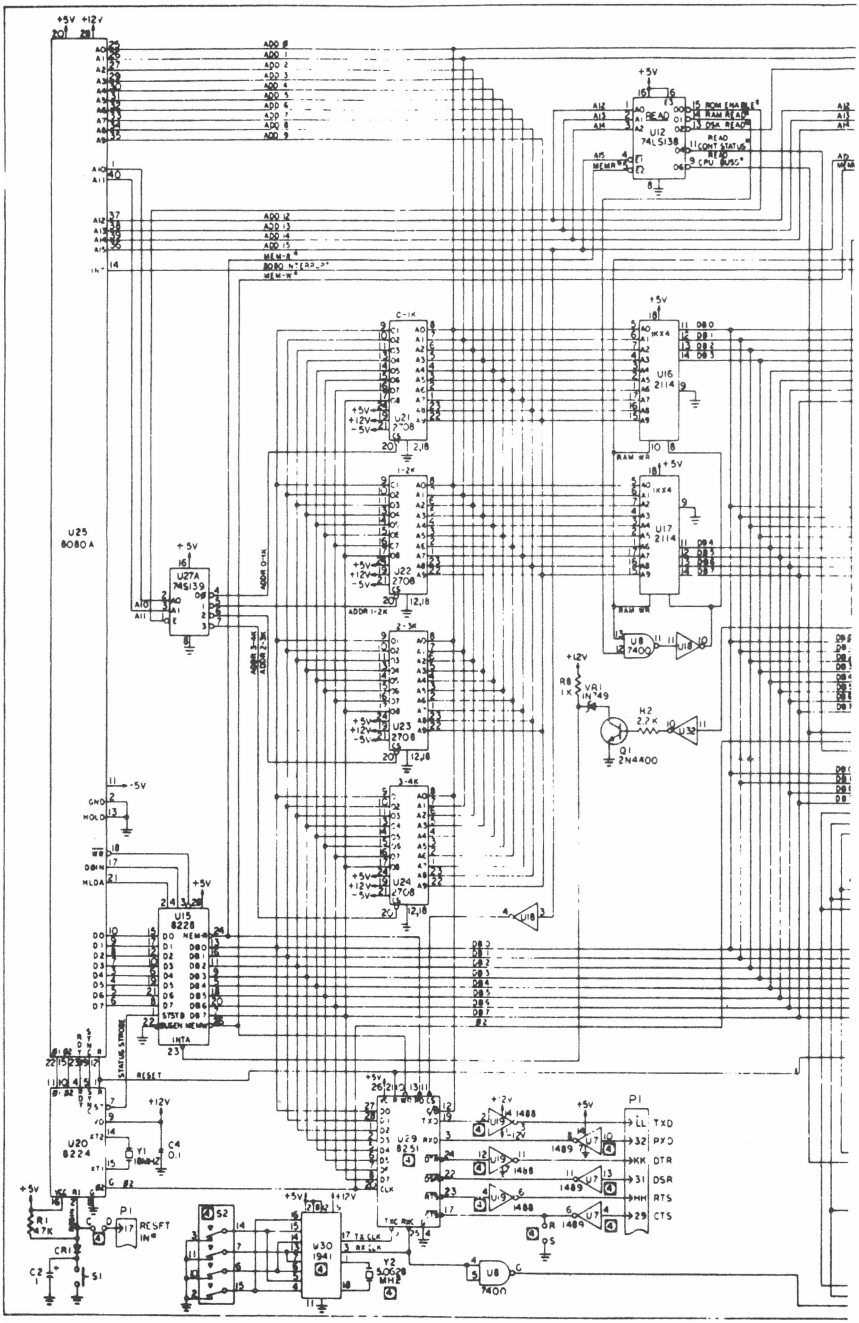


Fig. 4-116: PROM e CPU

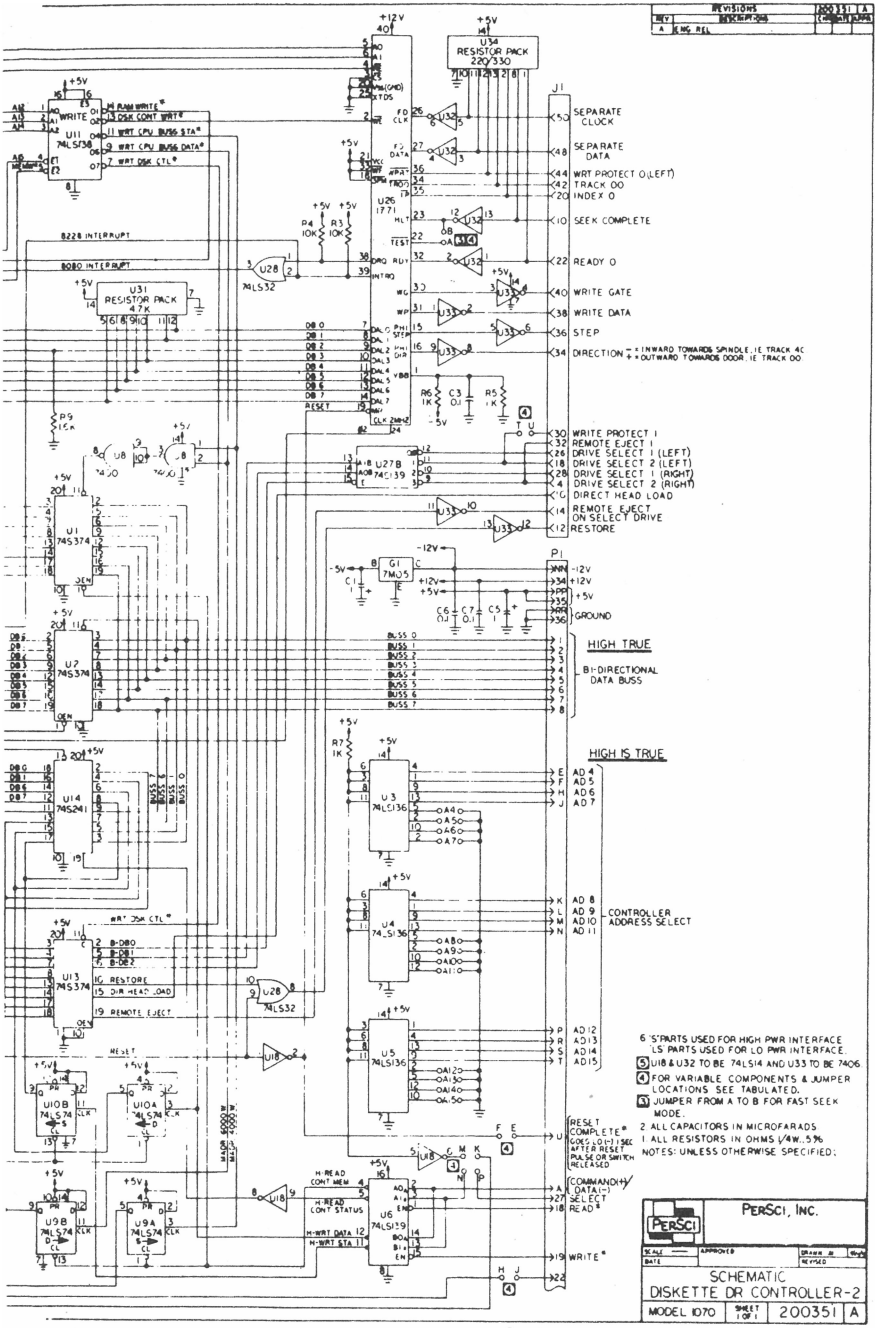


Fig. 4-117: Controllore, memorie temporanee

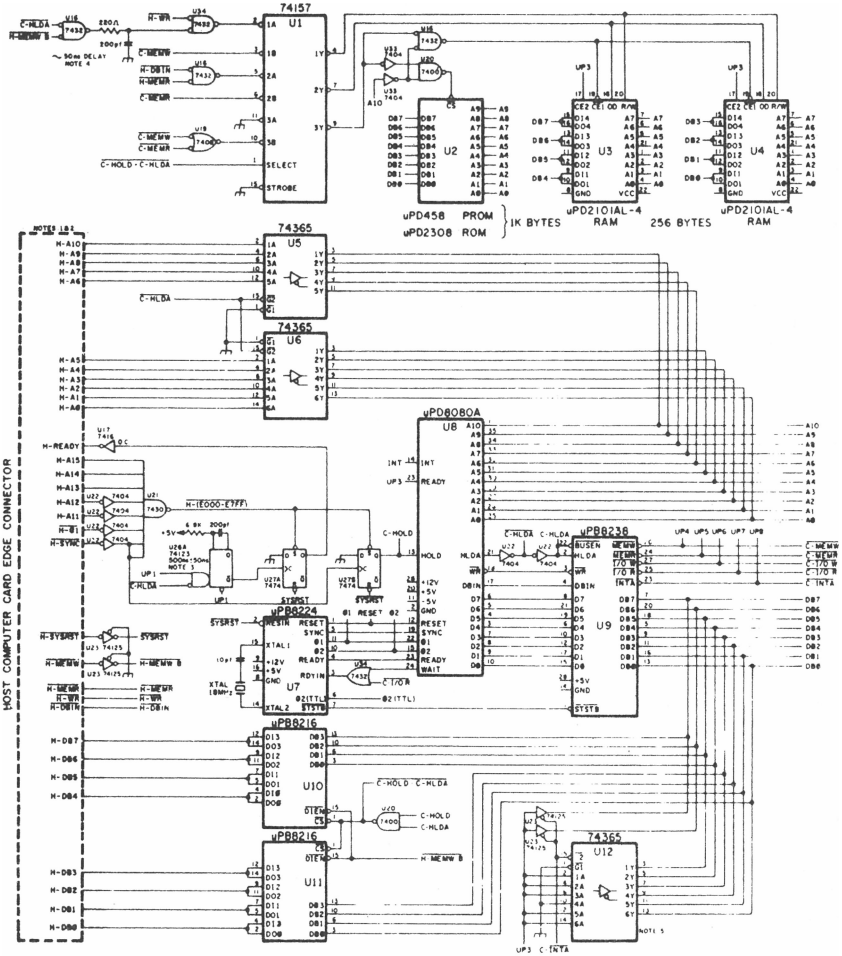


Fig. 4-118: Controllore di disco con 8080 della NEC

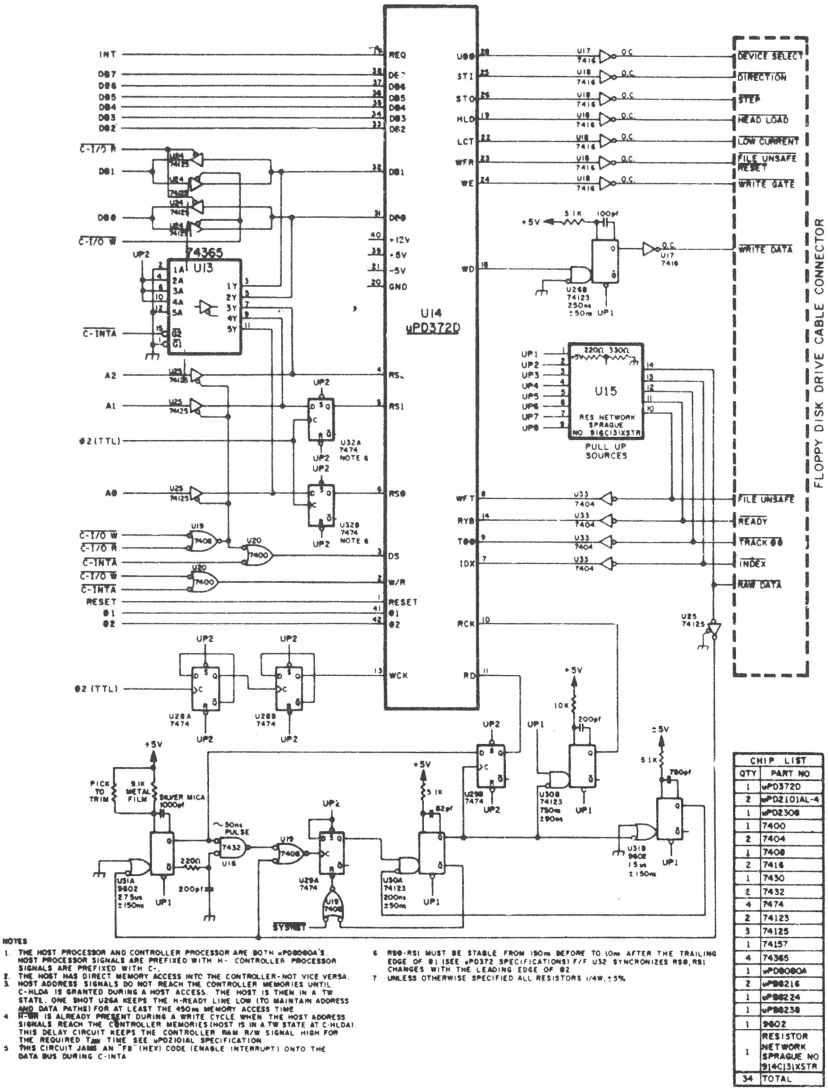


Fig. 4-119: Controllore di disco con 8080 della NEC

SEZIONE V: CONTROLLORE DI DISCO FLOPPY NEC

Il modulo FDC della NEC ha il nome UPD372D. Esso è compatibile con il 3740 IBM così come con il mini- floppy della SHUGART. Esso presenta le caratteristiche comuni, come la generazione del CRC, l'impulso di passo programmabile, velocità di salto di traccia, dimensione di settore, velocità di trasferimento dati. Inoltre esso controlla fino a 4 pilotaggi di disco, ma con lettura/scrittura limitata ad un solo pilotaggio, e simultanea ricerca di traccia sugli altri.

Altre unità di pilotaggio di disco sono:
CAL COMP 140, CDC BR 803, GSI 050 e 110, INNOVEX 210, ORBIS 74, PERSCI 75, PERTEC FD400, POTTER DD4740, SYCOR 145.

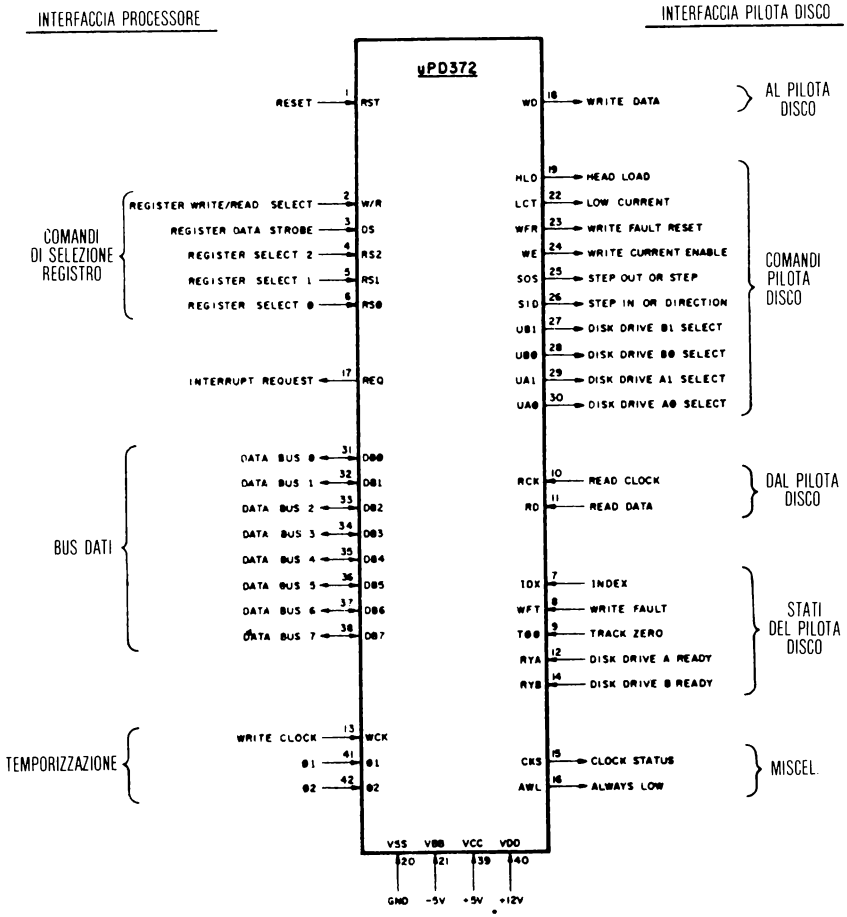


Fig. 4-120: FDC UPD372 della NEC

SEZIONE VI: FDC 6843 MOTOROLA

Il modulo FDC è progettato per interfacciare direttamente il 6800. Esso presenta 10 macrocomandi:

1. Ricerca traccia 0 (STZ)
2. Ricerca (SEK)
3. Scrittura di singolo settore (SSW)
4. SSW con indicazione di indirizzo con dati cancellati (SWD)
5. Lettura di singolo settore (SSR)
6. Lettura CRC (RCR)
7. Scrittura su più settori (MSW)
8. Lettura su più settori (MSR)
9. Scrittura in formato libero (FFW)
10. Lettura in formato libero (FFR)

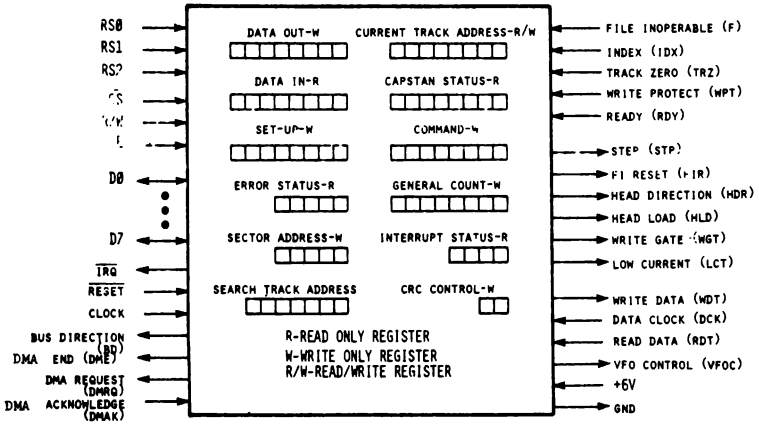


Fig. 4-121: Formato dei registri

In figura 4-122 è indicato il programma in linguaggio assembler per il pilotaggio del 6843 mediante il processore 6800. Il sistema richiede l'uso di un controllore DMA. Le routines opereranno la scrittura e la lettura sul disco floppy attraverso il 6843 utilizzando i registri indicati in figura. È da notare la indicazione esplicita di dove sono tali registri, e i loro indirizzi nello spazio di I/O. Routines addizionali di errore e di gestione del file devono essere aggiunte a questo programma per realizzare un «package» completo di software per il pilotaggio di disco floppy.

```
                ROUTINE TO WRITE TO, AND READ FROM
                ONE SECTOR OF THE FLOPPY DISK
                USING THE MC 6843 FDC
                ORG                 $0000

                FDC REGISTERS

DORREC EQU      $DFF0       W/O WRITE TO FLOPPY
DIRREQ EQU      $DFF0       R/O READ FROM FLOPPY
CTAREQ EQU      $DFF1       R/W CURRENT TRACK
CMRREQ EQU      $DFF2       W/O MACRO COMND & IRQ MASK
                            STRB MASK; WRT CLK CNTRL;
                            DMA ENABLE

ISRREG EQU      $DFF2       R/O IRQ REG. B3=I FROM STRB,
                            B2=STAT SEN. REQ., B1=STC.,
                            B0=MAC COM END.

SURREG EQU      $DFF3       W/O B0–3=ST TM; B4–7=SEK TM
STAREG EQU      $DFF3       R/O STRA 0–DTR; 1–DEL
                            DATA DETECT; 2–READY;
                            3–TRZ; 4–WPT; 5–TRK NOT EQUAL
                            6–INDEX; 7–BUSY;
SARREG EQU      $DFF4       W/O START SEC ADD DB, 0–4

STBREG EQU      $DFF4       R/O STAT B; 0–DTE; 1–CRC;
                            2–DTA MK NO DET; 3-NO SEC ADD;
                            4–SEEK ERROR; 5–FILE INOP;
                            6–WRITE ERROR; 7–HARD ERROR;

GCRREG EQU      $DFF5       W/O GEN CNT REG;
                            TRK FOR SEK; ALSO SECT CNT
                            FOR MULTI SECT R/W

CCRREG EQU      $DFF6       W/O SET CRC CONTROL
                            DB0=CRC SHIFT
                            DBI=CRC ENBLE
                            THIS IS FOR FREE FORMAT R/W

LTAREG EQU      $DFF7       W/O TRK NO FOR TRK ADD SRCH
```

Fig. 4-122: Linguaggio Assembly del driver 6843

```
                                    TO COMPARE TO ID FLD ON DSK
FDSELI  EQU     $DFF8               FL DSK SELECT ADD

                ADDITIONAL EQUATE STATEMENTS USED

XBPRNT EQU      $F728               X-BUG LOC TO START PRINTNG
BEGADD EQU      $FFOR               X-BUG BEGIN PRINT VECTOR
ENADD  EQU      $FFOC               X-BUG END PRINT VECTOR
IRQVEC EQU      $FFF3               X-BUG IRQ VECTOR

                IMAGE STORAGE FOR  FDC RESISTERS

DORIMG RMB      1                   W/ FLOP GETS DATA TO WRT HR
DIRIMG RMB      1                   R/ FROM FLOPPY
CMRIMG RMB      1                   MACRO COMM REG IMAGE

ISRIMG RMB      1                   INTERRUPT IMG
SURIMG RMB      1                   SETUP REG IMAGE
SARIMG RMB      1                   SECTO ADD IMG
STBIMG RMB      1                   STATUS REG B IMG

GCRIGM RMB      1                   GEN COUNT REG IMAGE
CCRIMG RMB      1                   CRC CONTROL REG IMAGE
SELIMG RMB      1                   DISK SELECT IMAGE

                STATUS FLAG BUFFERS

RSTKPR RMB      2                   STR RCV DATA BUFF PTR
STKPTR RMB      2                   STR STK PTR IF DO PSH OR PULL
INXSTR RMB      2                   STORE THE INDEX REG HERE
SSRFLG RMB      1                   FLAG IF WE FIND STATUS SENSE

TOTSEC RMB      1                   TOTAL SEC TO BE R/W
STRPRT RMB      2                   START ADD TO PRINT A BUFFER
ENDPRT RMB      2                   END ADD OF BUFFER FOR PRINT

TRKNUMRMB       1                   TRACK NUMBER FOR LTAR & GCR
SECNUM RMB      1                   SECTOR NUMBER FOR SAR
```

Fig. 4-122: Continuazione

```
                ORG $0000
          DATA BUFFER FOR READ DATA

REDBUF RMB    80        SAVE DEC 128 LOC FOR 1 SEC
               OF READ STORAGE BUFFER

               ORG                 $100
       PROGRAM AND FLOPPY DISK INITIALIZE

       SEI              SET THE INTERRUPT MASK
FLZERO CLR    X
       DEX
       BNE    FLZERO    LOOP UNTIL DONE

CLRMEM LDX    #$00FF    CLR RD DATA STORAGE BUFF
       STX    RSTKPR    USE AS RECV DATA PTR
MEZERO CLR    X         CLEAR THE RECV BUFFER
       DEX
       CPX    #$0080    ARE WE TO ADDRESS 80
       BNE    MEZERO    IF = 0 GO ON
NXTVEC LDX    #$0080    SETUP BEG PRINTOUT VECTOR
       STX    STRPRT    FOR EXBUG PRINT
       LDX    #$00FF    SETUP END PRINTOUT VECTOR
       STX    ENDPRT    FOR EXBUG PRINT
       LDA A  #$80      LOAD DATA FOR DOPREG
       STA A  DORIMG    SO WE HAVE DATA TO WORK WITH
       LDA A  #$03      SELECT FD #0
       STA A  FDSELI
       STA A  SELIMG
       LDA A  #$30      SET FOR TRACK 30
       STA A  TRKNUMWE WILL START WITH THIS TRACK
       LDA A  #$64      SETUP SEEK & SET TIMES
       STA A  SURREG    SEEK=6MS=IM6
       STA A  SURIMG    SETL=16MS=4X4
       LDA A  #$5
       STA A  TOTSEC    TOTAL SEC IN MULSEC R/W
       STA A  SECNUM    ADD OF FIRST SECT TO BE READ
       LDS    #$0FFF    SET STACK POINTER FOR LOTS OF
                        STORAGE AREA IF NEEDED
```

Fig. 4-122: Continuazione

```
        LDA A   STAREG
        BIT A   #$04    IS DRIVE READY?
        BNE     STZAGN  GO TO STZ IF WE ARE READY
        SWI             READY NOT THERE RETURN

                        ORG $0400
        SUBROUTINE TO REMOVE DATA TRANSFER
ERRORS OF PREVIOUS OPERATION FROM REGISTERS

CLRERR INC      DIRREG  REMOVE DIR FROM STRA
        TST     STBREG  REMOVE DTE FROM STRB
        TST     ISPREG  CLEAR ISR
        TST     STBREG
        BEQ     DONEOO
        SWI             LEAVE IF STRB WILL NOT CLR
DONEDO RTS              RETURN NO ERRORS FOUND
        END
```

Fig. 4-122: Conclusione

Esso è naturalmente equipaggiato con due ritardi programmabili per il tempo di ricerca e per il tempo di assestamento. I segnali del modulo sono indicati in figura 4-65. Questo modulo FDC richiede tre canali DMA. Esso usa una media del tre per cento del tempo di MPU. Assumendo una velocità di trasferimento di 256 KPS, il carico massimo dello MPU è pari al 12,5%.

SEZIONE VII: CONTROLLORE ROCKWELL DI DISCO FLOPPY

In figura 4-123 sono indicate le interconnessioni fondamentali di tale FDC ad un sistema Rockwell. Esso usa tre canali DMA (vedi figura 4-124), dei quali il canale 7 ripristina il canale 1. Le istruzioni di I/O del modulo FDC sono indicate in figura 4-125. Le routines tipiche per disco floppy relative al Rockwell PPS-8 sono indicate in figura 4-126.

FDC INTELLIGENTE

Un «FDC intelligente» è tale se rende il disco trasparente all'utente. Le caratteristiche di un controllore di disco floppy «intelligente» possono includere quanto segue:

- indirizzamento simbolico del file (richiede un sistema operativo basato sul disco – Disk Operating System).

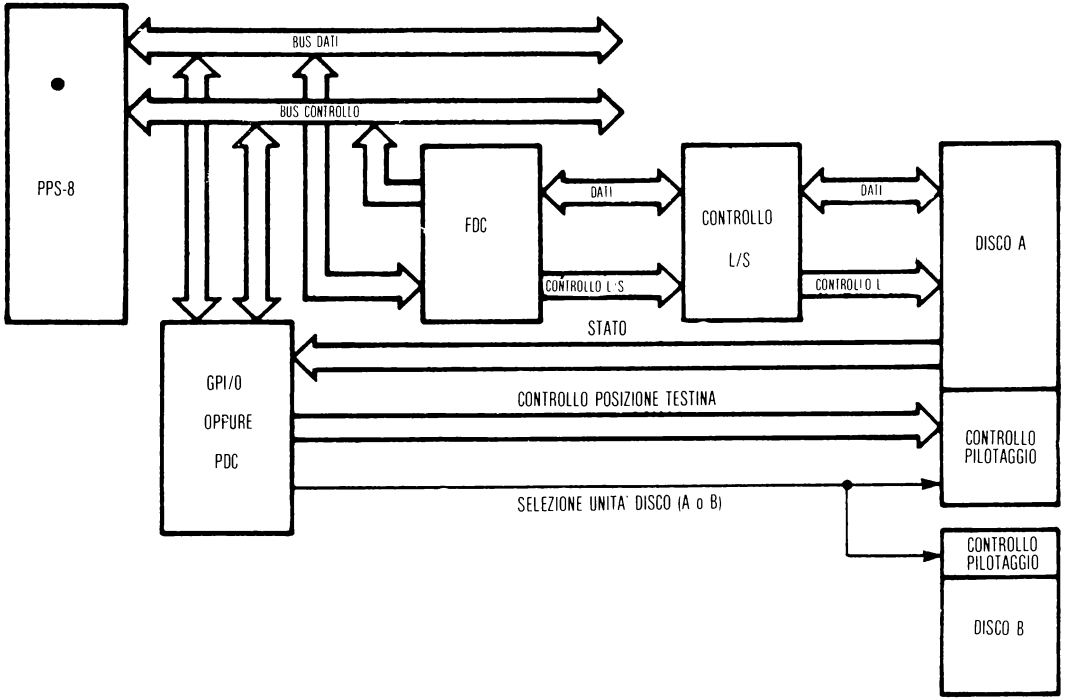


Fig. 4-123: FDC realizzato con PPS-8

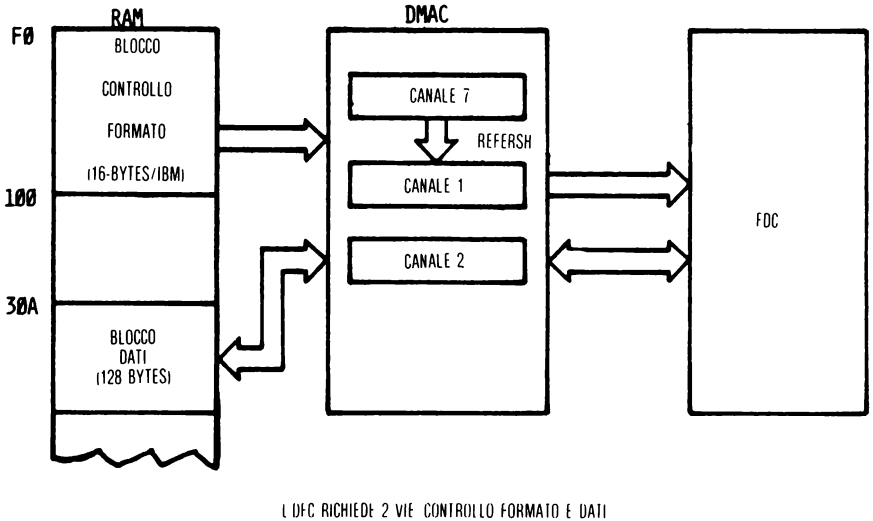


Fig. 4-124: Diagramma a blocchi del DMAC dell'FDC con PPS-8

01SS0000	NON OPERATIVO
01SS0001	START
01SS0010	IMPEGNO
01SS0011	AZZERAMENTO
01SS1010	DATI PRONTI
01SS1100	STATO LETTURA
01SS1101	STATO LETTURA
01SS0100	NON OPERATIVO
01SS1110	NON OPERATIVO
01SS100	LETTURA NON DEFINITA
00001000	STATO DI INTERRUZIONE LETTURA

Fig. 4-125: Comandi

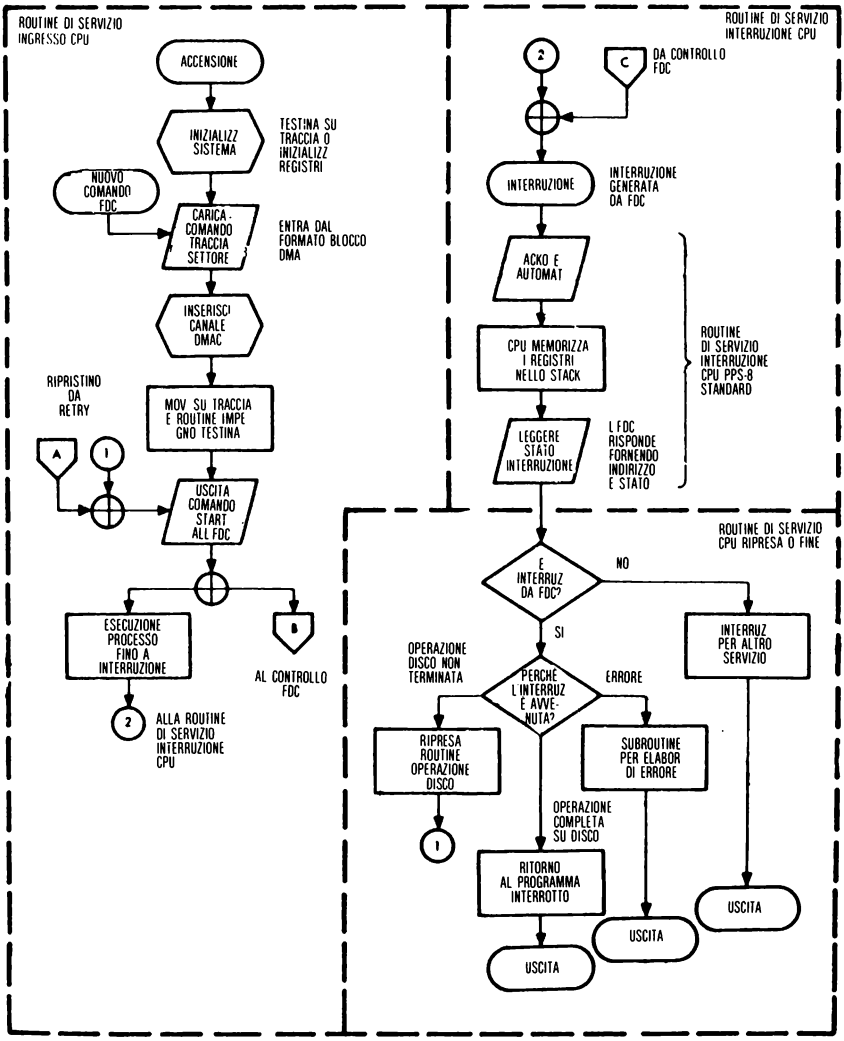


Fig. 4-126: Diagramma di flusso software

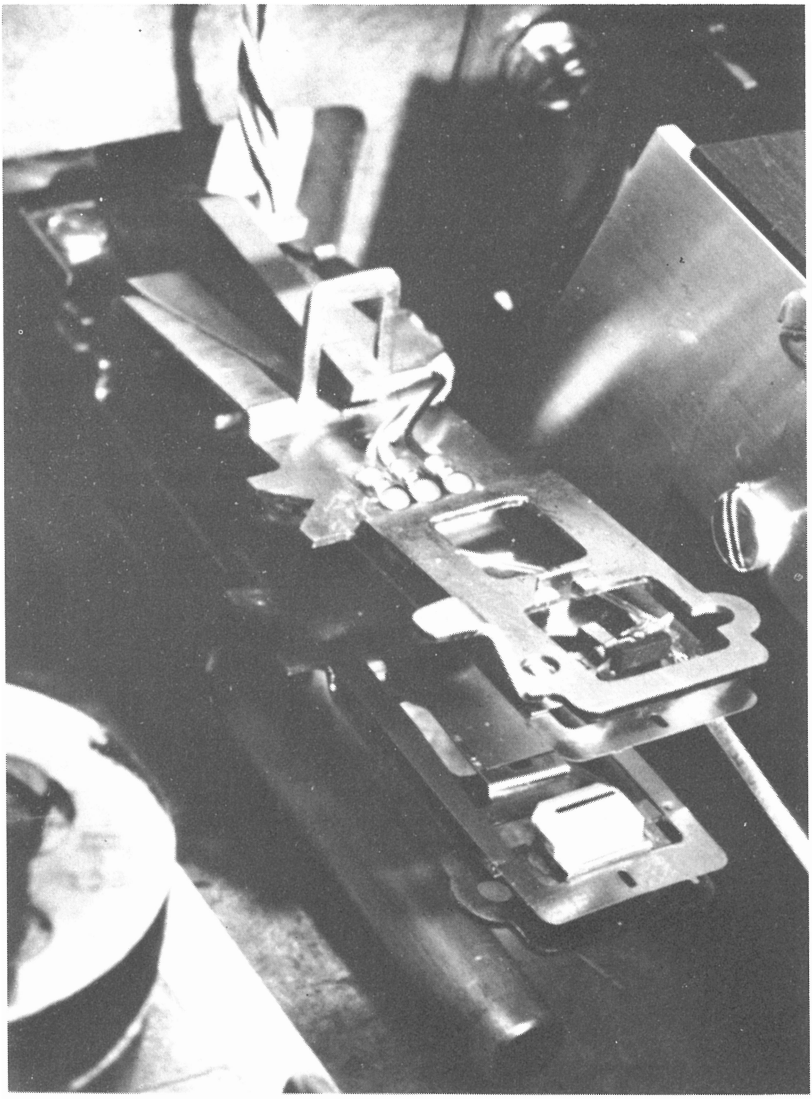


Fig. 4-127: Dettaglio di una testina a doppia faccia (Shugart)

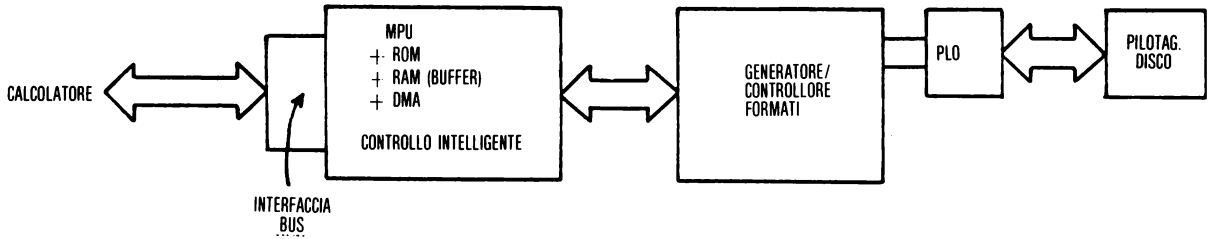


Fig. 4-128: Un controllore intelligente di floppy

- allocazione automatica dello spazio sul dischetto (richiede un sistema di gestione dei file – File Management System)
- gestione della testata del file con la data della creazione, la data della ultima modifica.
- gestione della traccia indice
- composizione del file
- memorizzazione tampone di ingresso - uscita
- diverse interfacce opzionali (RS232,S-100,8 bit parallela)
- inizializzazione del formato del dischetto (spazi, indicatori, campi di identificazione, campi dati)
- interallacciamento di settore
- gestione guida («directory») di file
- indicazione di spazio (raccolta di aree inutilizzate)
- assegnazione/annullamento di nomi
- diversi metodi di accesso: sequenziale, casuale, diretto, stringa
- copiatura di file
- rivelazione di errore e auto ripristino dopo errori software
- diagnostica

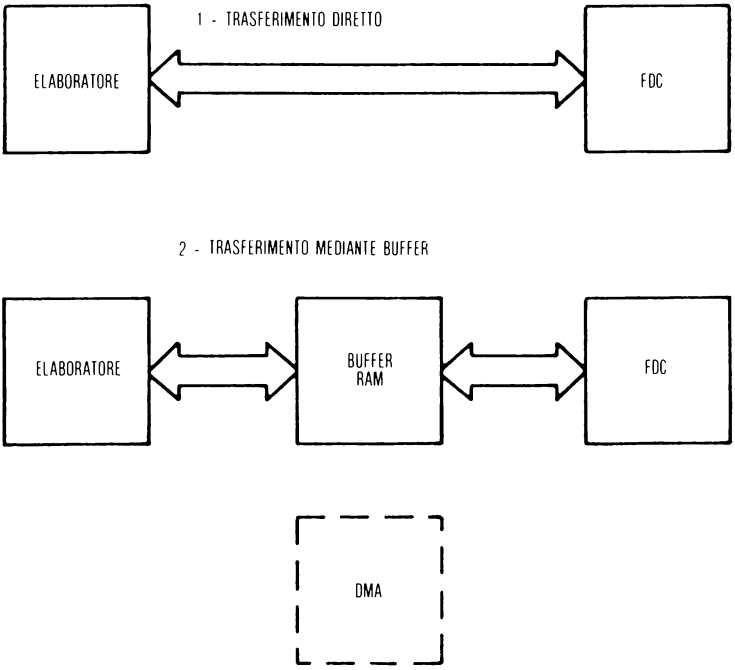


Fig. 4-129: Comunicazione con un FDC intelligente

- specifica di codice
  - scrittura ASCII, HEX (esadecimale), dati cancellati
  - lettura ASCII, HEX
- memorizzazione temporanea del contenuto di uno schermo
- caricamento su memoria temporanea.

In pratica, un FDC intelligente, realizza tali funzioni in software, e include una MPU nella piastra, ad esempio un 8080 o un 6800.

Per fornire una memorizzazione temporanea dei dati è normalmente disponibile una memoria con doppia porta. Una dimensione tipica della memoria tampone è 256 bytes per il pilotaggio del disco. In una configurazione tipica ci sarà un K di memoria per un numero di pilotaggi fino a quattro.

Il FMS (File Management System) e il DOS (Disc Operating System) sono residenti in ROM per efficienza. Dimensione tipica è 4K byte. Un controllore completamente intelligente richiede soltanto un *minimo* pilotaggio dell'I/O verso la CPU centrale (meno di 256 bytes è la memoria tipica).

La traccia indice del disco (traccia 00) è usata per contenere i riferimenti degli indici dei file (tipicamente 100). È essenzialmente usata come una tabella dei contenuti. La sua prima sezione è un volume di ID. Gli altri settori contengono i riferimenti dell'indice del file.

Le informazioni tipiche contenute in un *riferimento di indici* contengono:

- il nome del file, il numero della versione
- il numero di pilotaggio
- il tipo di file
- l'indirizzo di partenza, e lunghezza
- la posizione dell'indicatore EOF
- la data della creazione (= età del file)
- la data dell'ultimo aggiornamento

Gestione della memoria

I file sono normalmente una sequenza di settori per i quali è garantita la adiacenza in modo che essi possono essere scritti e letti alla massima velocità del disco. Tuttavia, quando un file è cancellato, il processo di disallocazione lascerà spazi nel disco. Spazi non utilizzati consumano memoria perchè la dividono in sezioni. Pertanto periodicamente, o quando non può essere garantita una richiesta di memorizzazione, risulta necessario compattare gli spazi. Il processo è conosciuto come «garbage collection», o compattazione.

Il Sistema Gestione File (FMS)

Un FMS ha la funzione di rendere la reale gestione della memoria invisibile all'utente. Un FMS tipico fornisce le seguenti funzioni, o comandi:

- alloca un file di n settori con il-nome-
- cancella il file
- rendi il file disponibile, disattivalo («open/close»)
- copia
- R/W in vari formati
- cambia il -nome- con il -nome-
- descrittore di lettura/modifica
- test di diagnostica

Interfaccia di controllore intelligente

Un controllore intelligente richiede una interfaccia minima verso il calcolatore, e nessuna interfaccia verso il pilotaggio del disco.

L'hardware interfaccia verso l'elaboratore tipicamente mediante la RS232, la S100 o mediante 8 bit in parallelo (attraverso la memoria, per una indipendenza del processore) (vedere il capitolo 6 per la descrizione della RS232 e della S100).

Metodi di accesso al file

1. Sequenziale, o a stringa
   Si accede e si memorizza l'intero file in modo continuo. Ciò implica l'uso di settori contigui. Il metodo è semplice ed efficiente se tutti i dati devono essere letti o scritti.
2. Lunghezza variabile, o mediante puntatori.
   Un file è gestito come una sequenza di record di lunghezza variabile. È usata quando porzioni di un file saranno modificate.
3. Accesso casuale, o relativo
   Qualunque byte (o numero di bytes) può essere scritto/letto, anche attraverso delimitazioni di settori.
4. Accesso diretto
   Qualunque settore o traccia può essere direttamente specificata, mediante baypass del sistema di gestione dei files.

Riferimento sul formato dei dischetti

IBM Diskette for Standard Data Interchange, CA 21 - 9182-0, File No. GENL 03180.

SINTETIZZATORE MUSICALE

Una delle più importanti applicazioni delle interfacce in esame è il progetto hardware del sintetizzatore musicale. È di seguito svolta una breve descrizione del suo funzionamento. Esso consiste di due canali: il sinistro e il destro. Ciascun canale copre un campo di tono musicale leggermente diverso dall'adiacente. Essi corri-

spondono approssimativamente a quelli coperti dalle mani sinistra e destra nel piano. Tutti i toni che si possono ottenere in un piano, eccetto che i 7 tasti più bassi, sono ottenibili mediante tali sintetizzatori. L'hardware sarà esaminato attentamente nelle sezioni successive nelle seguenti sue parti: il DCO, il generatore armonico, il circuito di intonazione, l'articolatore e il generatore di tempo. Con la eccezione delle differenze nei campi coperti, i due canali sono identici. È qui descritto, pertanto, un solo canale.

Il DCO

La base di ogni sintetizzatore musicale è l'oscillatore che determina la frequenza o il tono della nota ascoltata.

In questo sintetizzatore musicale l'entità controllante è un microprocessore basato su un microcalcolatore, i cui ingressi e le cui uscite sono strettamente digitali per loro natura. Un approccio digitale per generare le frequenze necessarie è quello qui usato. Usando un programma di temporizzazione, il microcalcolatore potrebbe generare le frequenze stesse; questo, tuttavia, limiterà il numero di operazioni che il micro può realizzare, sottoutilizzando la sua potenza come unità di calcolo. Pertanto è stato sviluppato un *oscillatore controllato dai dati* esterno (DCO). Come il

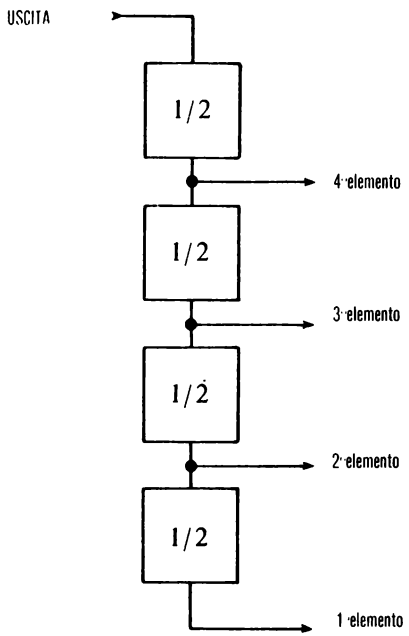


Fig. 4-130: Catena di divisione

nome indica, la frequenza della sua uscita è determinata dal dato digitale presentato al suo ingresso.

Il DCO è essenzialmente composto da tre contatori in cascata binari a quattro bit e preposizionabili. Esso opera come segue: un numero di 12 bit è fornito al DCO dal microprocessore. Il contatore binario risulta così preposizionato con questo numero di 12 bit, e successivamente conta a partire dal valore dell'ingresso dell'oscillatore.

Quando il contatore raggiunge il suo massimo conteggio, è prodotto un impulso di riporto. Tale impulso è usato come uscita e per preposizionare il contatore nuovamente al precedente numero di 12 bit. Pertanto, tanto più grande è il valore di tale numero a 12 bit, più presto il contatore raggiunge il suo massimo conteggio, e più spesso è prodotto un impulso in uscita; risulta una frequenza di uscita più alta. Se N rappresenta il complemento a due del numero a 12 bit, la frequenza di uscita risultante è:

$$f_{out} = OSC/N$$

È da notare che N può avere solo valori interi. Questo limita la precisione con la quale può essere approssimata, una certa frequenza. È stata fatta un'analisi per de-

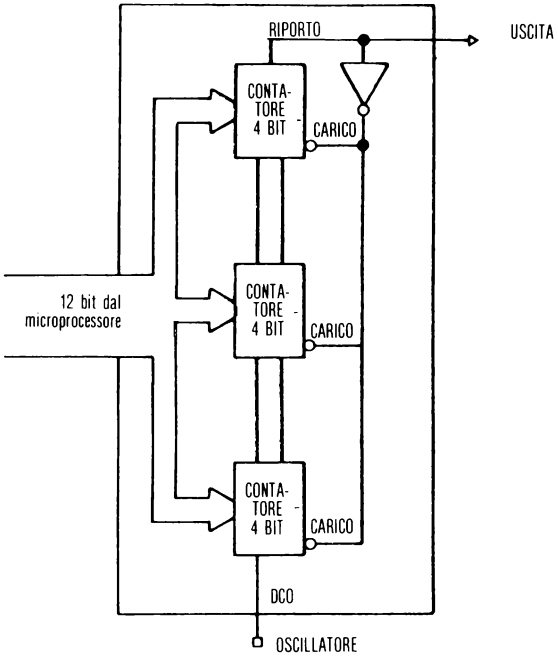


Fig. 4-131: Il DCO

terminare se 12 bit permettono una precisione sufficiente per il sintetizzatore musicale.

Per un oscillatore a 5MHz è risultato che, per un campo di 4 ottave, il massimo errore possibile è 2,25 centesimi, dove un centesimo è 1/100 della distanza logaritmica tra due qualsiasi note adiacenti (mezzo passo). Ciò è abbastanza accettabile. Conseguentemente, questo DCO a 12 bit è stato usato per generare le frequenze base del sintetizzatore.

Il generatore armonico

Se l'uscita del DCO fosse fornita all'ascolto attraverso l'altoparlante, essa avrebbe una qualità del timbro poco gradevole. Ciò è dovuto alla scarsa presenza di armoniche. Armoniche di una nota sono le frequenze multiple intere della fondamentale. La particolare caratteristica sonora che offre il suono di uno strumento musicale è la sua struttura armonica. Per variare il contenuto armonico è usato un gruppo di onde quadre il cui periodo sia sempre la metà della componente precedente, a partire dalla frequenza fondamentale.

Per produrre tali funzioni, l'uscita del DCO è iterativamente diviso per due usando flip-flop di tipo D

Usando i primi 4 membri del gruppo di divisione può essere approssimata qualunque forma d'onda, anche se l'errore può essere grande. Dopo avere esaminata

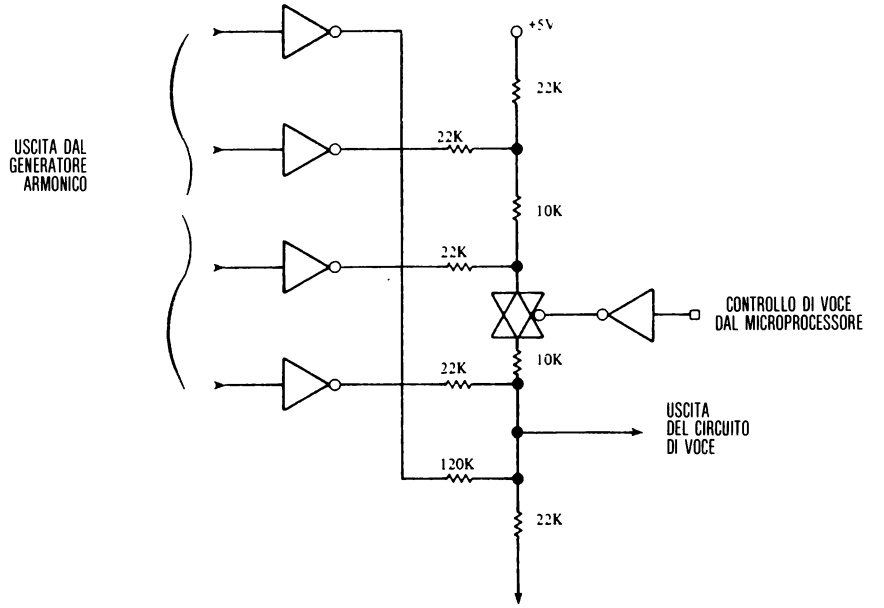


Fig. 4-132: Circuito di voce

la forma d'onda di un arpicordo e del piano è risultato che la più comune approssimazione alle forme d'onda di tali strumenti è quella di un'onda a dente di sega e di una onda quadra distorta.

L'uscita del generatore armonico non è adatta per essere trattata in modo da produrre questa forma d'onda poichè i livelli di uscita variano in ogni caso da 2,2 volt a 3,8 volt. Pertanto tali uscite sono memorizzate e incrementate di livello mediante CMOS, poichè la famiglia in logica CMOS presenta livelli di uscita che differiscono di poco dalla tensione di alimentazione. L'elemento poco comune che è presente in figura 4-132 insieme alla rete di resistenze è chiamato commutatore analogico bilaterale. Quando la sua tensione di ingresso di controllo è alta, il commutatore analogico si comporta come una resistenza di 200 ohm per tutte le tensioni pesate dalla tensione di alimentazione. Quando la tensione di controllo di ingresso è bassa, il commutatore analogico si comporta come una resistenza di 200 Megaohm. Questo commutatore analogico è usato per selezionare una delle due voci.

Articolatore

Un altro fattore che dà colore alle caratteristiche di uno strumento musicale è la velocità di decadimento del suono. Per esempio, quando è premuto un tasto di un arpicordo, è generata una stringa la cui sonorità decade rapidamente. Diversamente nel piano, se il tasto è tenuto premuto, la sonorità della nota decade molto lentamente. Ma liberando il tasto, la nota si estingue. L'equivalente di queste funzioni è realizzato nel sintetizzatore mediante il circuito articolatore.

L'articolatore é essenzialmente un moltiplicatore analogico; uno dei suoi ingressi é l'uscita del circuito di voce; l'altro ingresso é l'inviluppo di decadimento. La funzione di moltiplicazione é praticamente realizzata controllando un commutatore analogico bilaterale CMOS in modo analogico. Per realizzare diversi inviluppi di decadimento, sono presenti vie addizionali di scarica su un condensatore. Le inserzio-

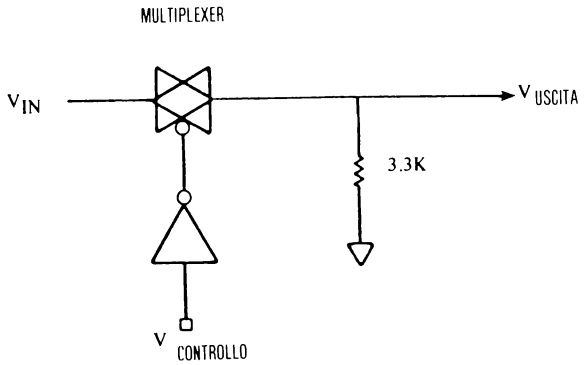


Fig. 4-133: Dettaglio del multiplatore dell'articolatore

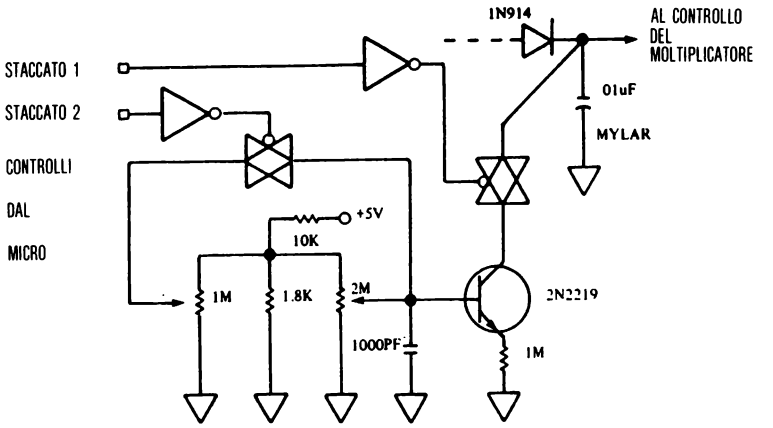


Fig. 4-134: Articolatore completo

ni di queste vie sono controllate dal microprocessore. Le velocità di scarica sono determinate dal potenziometro e possono essere regolate dall'utente per adattare il tasto relativo.

Generatore tempo

Nel generatore tempo é presente un oscillatore CMOS la cui frequenza é regolata da 0,25 Hz a 15 Hz. La frequenza di tale oscillatore determina la velocità con la quale la composizione procede da una nota all'altra. L'oscillatore é connesso ad un multivibratore monostabile, che produce un impulso di durata pari a 3 millisecondi sul fronte negativo del segnale di temporizzazione. Questo impulso é infine memorizzato temporaneamente e presentato ad una via di test del microprocessore. Questo impulso segnala al microprocessore che deve essere composta la nota successiva. Questo impulso è anche ritardato di 1,5 millisecondi prima di essere applicato al circuito articolatore. Questo dà sufficiente tempo al microprocessore per aggior-

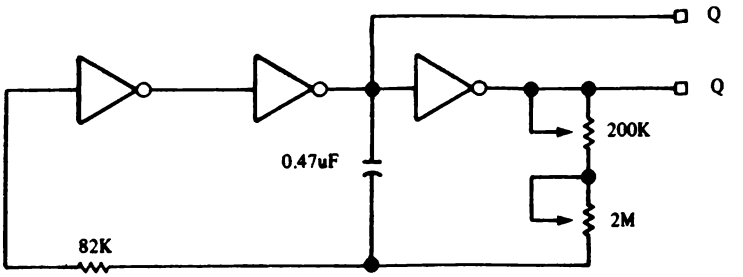


Fig. 4-135: Oscillatore tempo

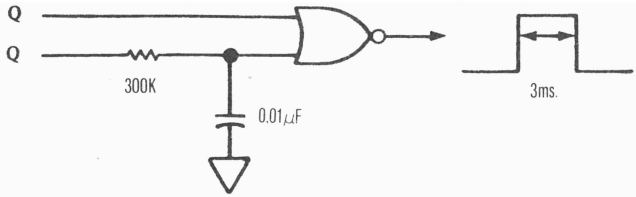


Fig. 4-136: Monostabile tempo

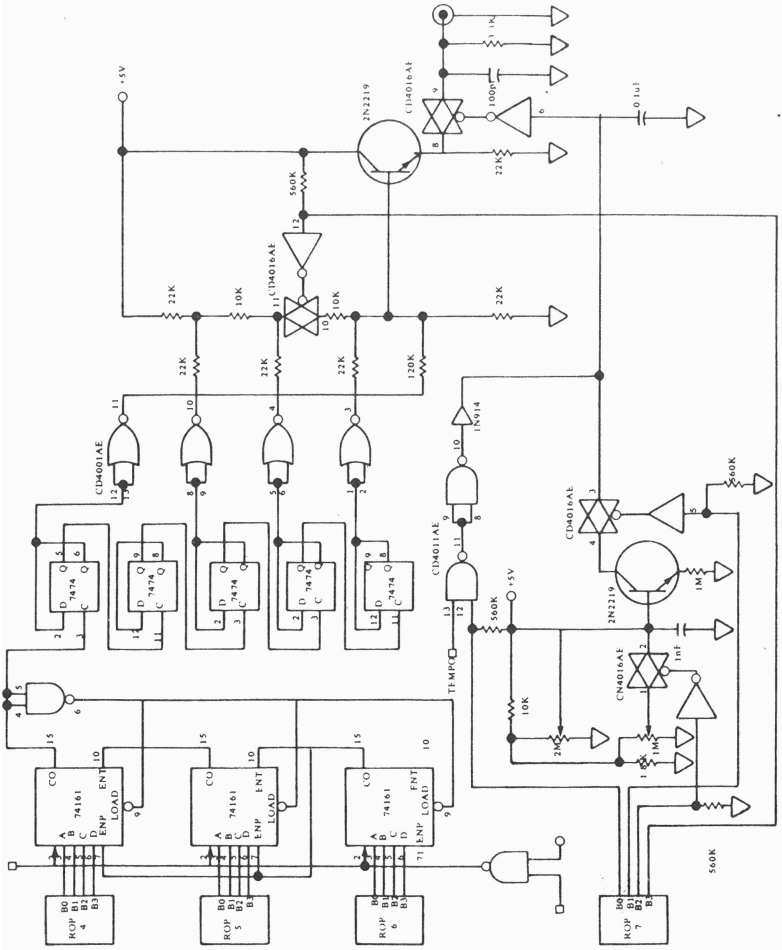


Fig. 4-137: Circuito di mano sinistra

nare i dati, da spedire al sintetizzatore musicale. Pertanto é comunicata al sintetizzatore la corretta tonalità, voce e la velocità di decadimento di ciascuna nota.

Il software

Per il sintetizzatore musicale il microprocessore gioca un ruolo di controllore, sequenziatore, e processore di dati. Fa seguito una breve descrizione funzionale: é richiamata da memoria una parola di una lista di codici musicali, mediante identi-

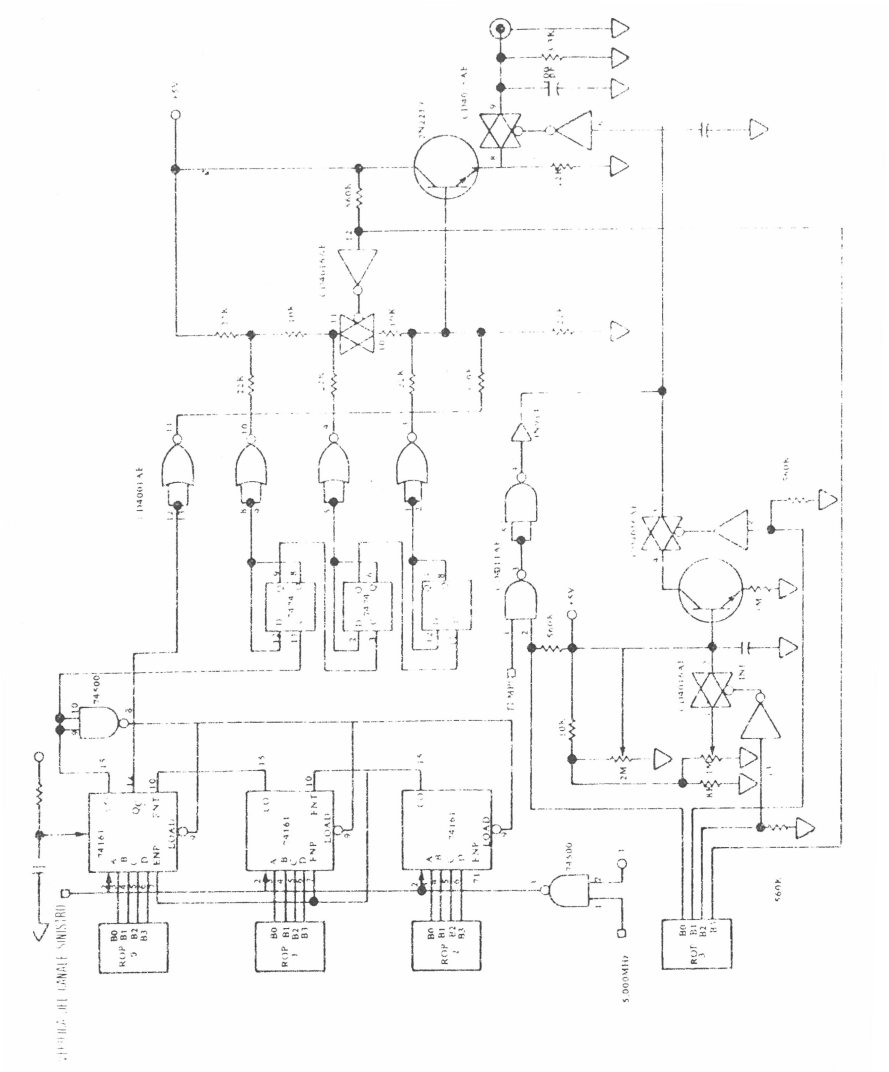


Fig. 4-138: Circuito di mano destra

ficazione di puntatore. Questo sarà: il codice per una pausa, per una delle 12 possibili note, per un cambiamento di ottava, o per un cambiamento di voce e articolazione. Se il codice corrisponde a nuovi valori di ottava o voce ed articolazione, è richiamata la parola successiva della lista ed é posta nella cella di memoria di controllo di ottava o voce. Se il codice corrisponde ad una nota, esso é decodificato per fornire il comando della nota specifica. Esso é presentato al DCO e pesato corrispondentemente al campo precedentemente assegnato. Sulla base della catena di divisioni già indicata risulta una nota o una pausa. Alla fine del conteggio é quindi disponibile l'informazione per suonare in un canale la nota successiva. La stessa procedura é ripetuta per l'altro canale. Il microprocessore resta successivamente in attesa di un segnale dall'hardware del sintetizzatore musicale (l'impulso del generatore, di tempo) che indica quando deve essere presentata in uscita l'informazione

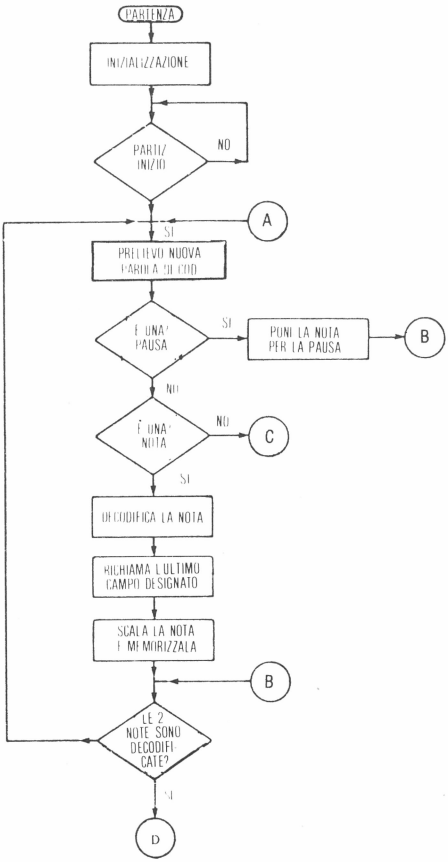


Fig. 4-139: Diagramma di flusso del ciclo principale

decodificata. Questa procedura continua per tutto il tempo della esecuzione di una composizione.

Le sequenze precedentemente indicate sono prodotte dal microprocessore attraverso l'esecuzione di un programma e un set di sottoprogrammi, o subroutine. Quanto di seguito indicato é una lista composta da: programma principale, subroutine di ricerca e prelievo, subroutine di pausa, subroutine di decodifica di nota, subroutine di scala corretta, la subroutine di sincronismo, la subroutine di trasmissione e quella di ripristino.

Il programma principale svolge la funzione di decodificare parole e di coordinare le subroutines con le parole decodificate. Esso determina anche quale mano deve essere decodificata nel momento specifico e successivamente, quale gruppo di celle

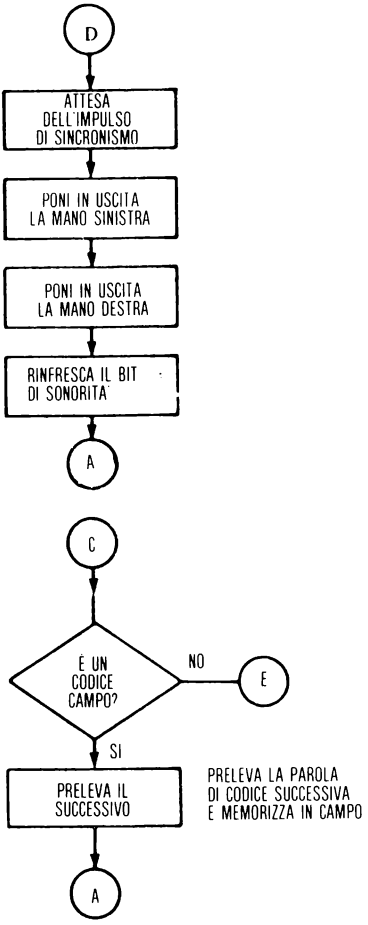


Fig. 4-140: Diagramma di flusso per la sonorità e il prelievo di codice

di memoria è attivo. A causa del continuo uso di subroutine è più facile comprendere il funzionamento del programma principale esaminando la struttura delle subroutine.

Quando la parola di codice richiamata dalla lista specifica una pausa, il programma principale pone il numero 4095_{10} nelle celle di memoria del DCO. Questo sarà interpretato dal DCO come la richiesta di una frequenza infinita. A causa della sua incapacità a produrla, il DCO smette di oscillare. Il programma principale chiama allora la subroutine di pausa, che azzera la porta del bit di sonorità della cella di memoria del controllo di voce.

Prima di descrivere le subroutine di decodificazione di nota e quella per mettere la nota nella scala corretta, occorre spendere altre parole sul DCO. Poiché la frequenza di uscita del DCO é data da

$$f_{out} = 5 \text{ MHz}/N$$

per generare le note é necessario memorizzare una tabella di valori di N, che corrispondono alle frequenze di tutte le possibili note.

Anche se questo metodo é certamente lineare l'individuazione delle singole note é una operazione abbastanza lunga. Una proprietà della musica della quale si trae

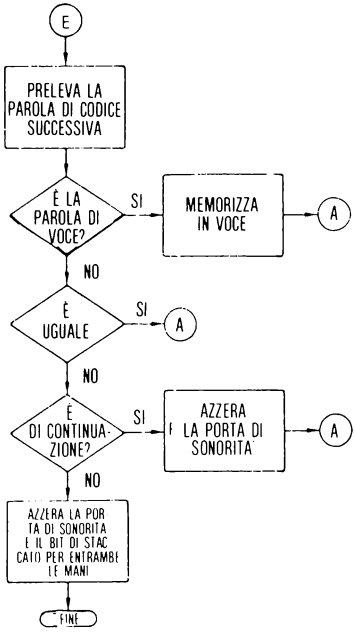


Fig. 4-141: Sequenze per prelievo, effetto voce, sonorità

beneficio é la ricorrenza delle note in ottave. Cioé la frequenza di una C media (261,81 Hz) é di valore doppio rispetto a quella di una C medio-bassa (130.81 Hz). Pertanto, se il valore N é noto per le note di una ottava, il valore relativo ad una nota in qualsiasi ottava é calcolabile moltiplicando o dividendo N per 2, operazione elementare per una macchina binaria. Ciò é quanto esattamente fanno le subroutine di decodifica di nota e quella per mettere la nota nella scala corretta. Quando il programma principale riconosce una parola di codice che specifica una nota sono chiamate tali subroutine e il numero N risultante é posto nella cella di memoria del DCO.

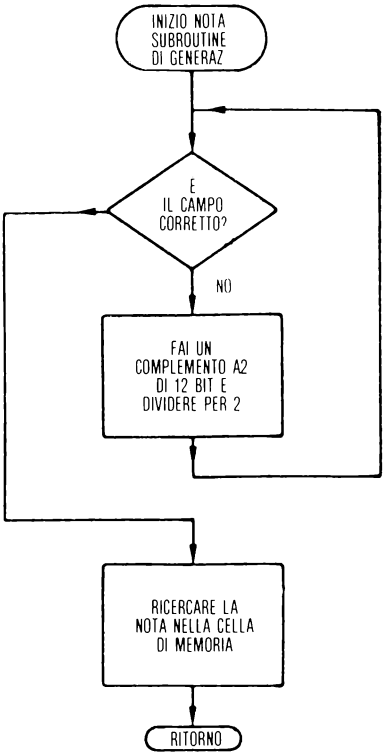
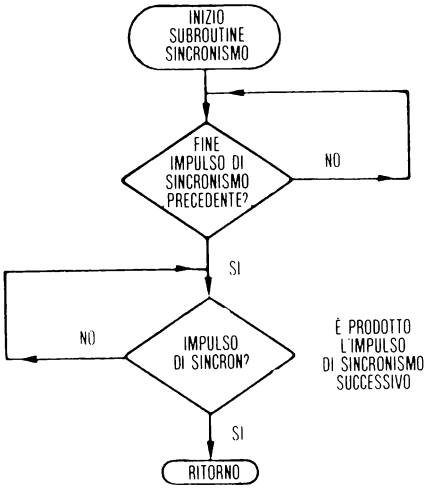


Fig. 4-142: Diagramma di flusso della subroutine di generazione a scala

Quando sono stati decifrati entrambi i canali, o mani, il programma principale chiama la subroutine di sincronismo che attende l'impulso generatore di tempo. Questo impulso indica al software che é tempo di spedire il gruppo successivo di informazioni all'hardware del sintetizzatore musicale. Il programma principale chiama allora la subroutine di trasmissione, che spedisce il contenuto della cella di memoria del DCO e di quella di controllo di voce al canale corretto. Ciò é fatto per



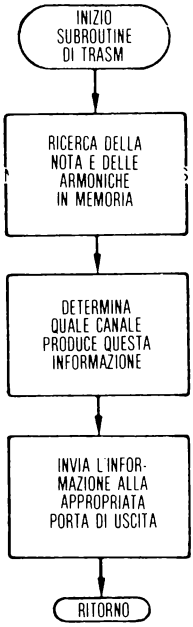


Fig. 4-143: Diagramma di flusso della subroutine di sincronismo e di trasmissione

Inventio 1 **J.S. Bach**

Musica Codificata per la Mano Destra

```
9.00,FF 09 BD 11 2D 0B D1 1D 09 D1 4F 49 F4 8F 49 F4
9:10,B4 68 96 84 BF 4D 24 F4 2F 44 F4 16 42 14 26 42
9:20,1D 1B 9D 21 D1 BD 22 1D 1B 98 69 8B 98 64 36 48
9:30,6F 4D 0B F4 D1 9F 4F 4B 86 43 14 36 48 69 8B 9D
9:40,21 D1 B8 BD 24 D1 8F 46 44 F4 00 00 00 04 68 96
9:50,34 3F 40 00 00 00 68 9B 89 68 F4 00 00 00 0B 98
9:60,69 8B 9F 40 00 00 00 D2 1D 1B 98 BA D2 1D 1B F4
9:70,AF 4B F4 D2 1F 42 F4 D1 6F 43 F4 AF 4B F4 3F 45
9:80,F4 6F 48 F4 9F 4B F4 F4 F4 F4 13 56 35 1D 21 D1
9:90,B9 D2 1D 1B 98 B9 D2 65 86 12 D1 B5 D2 21 D1 B9
9:A0,F4 86 6D 26 42 14 26 4F 4F 4F 4F 4F 4F 4F 4F 41
9:B0,24 62 41 2F 4F 4F 4F 4F 4F 4F 4F 44 21 D1 BD 22
9:C0,14 2F 4F 4F 4F 4F 4F 4F 4F 4D 1B D2 12 41 2D 1B
9:D0,D2 1F 4F 4F 4F 4F 4F 4F 4F 4D 19 BD 21 2D 1B D2
9:E0,1D 19 BD 21 24 62 41 24 68 96 84 9F 44 F4 1F 4D
9:F0,1B 99 76 42 64 76 89 1D 0B D1 92 89 F4 3F 36 CC
```

Fig. 4-144: Musica per la mano destra

```
8.00,FF 00 00 00 00 0D 19 BD  21 2D 1B D2 1D 19 D2 4F
8.10,4D 14 F4 00 00 0D 24 68  96 84 9F 43 F4 9F 4B F4
8.20,D3 1F 4D 24 F4 6F 48 F4  9F 41 F4 3F 44 F4 6F 48
8.30,F4 9F 4F 4F 4F 4D 1B D2  13 41 3D 1B D2 4F 4D 18
8.40,F4 9F 4B F4 D2 1F 43 F4  4F 41 F4 D1 8F 4F 49 BF
8.50,4D 0B F4 0D 14 68 96 84  BF 4D 24 F4 3F 44 F4 6D
8.60,1B D2 13 41 3D 1B D2 6F  4B F4 9F 4B F4 4D 34 21
8.70,D2 BD 32 14 2F 41 F4 2F  4D 2B F4 D3 16 42 14 26
8.80,4F 42 F4 4F 41 F4 27 64  26 47 64 21 D2 BD 32 14
8.90,21 D2 B9 8B 9D 31 D2 B9  86 58 69 8F 41 F4 BF 4F
8.A0,4D 31 D2 98 64 36 58 69  8B 9D 31 D2 BD 32 1F 4D
8.B0,26 F4 D3 1F 4D 21 F4 6F  4D 16 F4 00 00 D3 01 D2
8.C0,B9 8B AD 31 D2 BF 4F 4F  4F 4F 4F 4F 4F 46 89 B8
8.D0,96 8F 4F 4F 4F 4F 4F 4F  4F 4B 98 69 8B 9F 4F 4F
8.E0,4F 4F 4F 4F 4F 44 67 96  74 6F 47 F4 6F 44 F4 2F
8.F0,4B F4 9F 47 F4 6F 4D 32  F4 1F 4D 2B F4 3F 36 CC

A.00,D3 1D 1B D2 12 41 2D 1B  D2 1F 4D 19 F4 BF 4D 21
A.10,F4 2D 1B D2 12 4F 4D 14  F4 D0 9F 4F 4F C0
```

Fig. 4-145: Musica per la mano sinistra

ciascun canale. Infine, il programma principale chiama la subroutine di ripristino, che ripristina il bit della porta di sonorità nella cella di memoria di controllo di voce. Si ritorna, così, all'inizio del programma per il ciclo successivo.

Il diagramma di flusso é indicato nelle pagine successive.

Codifica della musica per il sintetizzatore musicale

Il software per questo sintetizzatore musicale é stato progettato in modo che l'accesso e l'esame continuo di una composizione musicale sia relativamente semplice e lineare. La musica codificata é trattata per l'azione di una singola mano alla volta.

Per ciascuna nota sono prima specificati il campo desiderato, la voce, i cambi di articolazione, e successivamente la nota, un codice di comando di sistema o una pausa. Se non sono presenti cambiamenti nel campo, nella voce, o nella articolazione, il campo trattato é cancellato. Il tipo di formato di ingresso é di seguito rappresentato.

La tabella di codice completa è indicata nelle pagine 218 e 219. Esse contengono la musica codificata relativa alle «First Two Part Inventions» di J. S. Bach.

I comandi di sistema sono trattati allo stesso modo delle note, nel senso che sono possibili solo piccoli cambiamenti rispetto alla nota precedentemente suonata. I comandi infatti indicano di continuare a suonare la stessa nota, o di ripetere la stessa nota, o ancora di smettere di suonare perché é terminata la composizione. Il comando di continuare a suonare la stessa nota significa che la nota sarà ancora ascoltata, ma non sarà più «toccata» una seconda volta. Ciò é ottenuto azzerando il bit della porta di sonorità nella cella di memoria di controllo della voce. Ripetere la stessa nota significa che non si deve fare alcun cambiamento: la nota é esattamente eguale a quella precedente.

AGGIORNAMENTO DELLA RAM DINAMICA

La RAM a MOS dinamica memorizza bit di informazione come cariche nelle capacità MOS. È richiesta una capacità per ciascun bit. Una operazione di lettura scarica la capacità e confronta la tensione rispetto ad una di riferimento. È necessaria un'operazione di riscrittura per mantenere il contenuto della RAM. Purtroppo, infatti, perdite entro il circuito MOS scaricano tali capacità in alcuni millisecondi. È necessario pertanto ripristinare la carica, generalmente ogni 2 millisecondi. Tale operazione é chiamata *ripristino (refresching)* della RAM. *Tutte le posizioni* entro la RAM devono pertanto essere ripristinate ogni 2 millisecondi.

Una RAM statica opera, invece, diversamente. Essa memorizza i bit in flip-flop. Essa non richiede clock, e conserva l'informazione fino a quando é applicata l'alimentazione. Tuttavia una cella RAM dinamica può essere realizzata con un singolo transistore MOS, comportando una maggiore densità. Tipicamente, una RAM dinamica é quattro volte più densa di una RAM statica, con un costo risultante minore. Le RAM dinamiche sono anche caratterizzate da un minore consumo di po-

tenza. Il loro svantaggio consiste nel fatto che richiedono un circuito di controllo del ripristino, spesso complesso. La RAM dinamica é spesso usata per memorie di maggiore dimensione (come 8K o 16K), mentre le RAM statiche tendono ad essere usate per sistemi più piccoli.

Per ridurre il numero di cicli di ripristino per un sistema di memorie RAM, una memoria tipica di 4K é organizzata in 64 righe e 64 colonne, richiedendo così soltanto 64 cicli di ripristino. La 2116, una RAM recente di 16K é strutturata in una matrice doppia, organizzata in 64 righe e 128 colonne. Poiché é possibile accedere contemporaneamente alle due matrici, sono necessari soltanto 64 cicli di ripristino.

Controllo del ripristino

Il circuito di controllo del ripristino ha il compito di ripristinare l'intera RAM entro 2 ms. Per accedere alla memoria sono utilizzate due tecniche fondamentali:

1 - *Modo in blocco:* il circuito di controllo ripristina le righe una alla volta, cioé in sequenza. Ciò é concettualmente semplice, ma rende la RAM indisponibile per il microprocessore per 64 cicli. Può essere facilmente calcolata la riduzione di disponibilità per il caso peggiore. Assumendo un clock di ciclo pari a 500 nanosecondi, e 64 cicli di ripristino entro 2 ms, sono perduti per ripristino 64 x 500 = 32 microsecondi. Tale valore ripetuto ogni 2 millisecondi rappresenta una perdita di 32/200 = 1,6%.

2 - *Modo distribuito o/a ciclo singolo*: il circuito di controllo di ripristino accede alla memoria ogni n microsecondi per ripristinare la riga successiva. Questa tecnica presenta una potenzialità di minori tempi di attesa per servizio del microprocessore, purché il numero dei bytes di sistema per gestire l'accesso in memoria sia mantenuto basso.

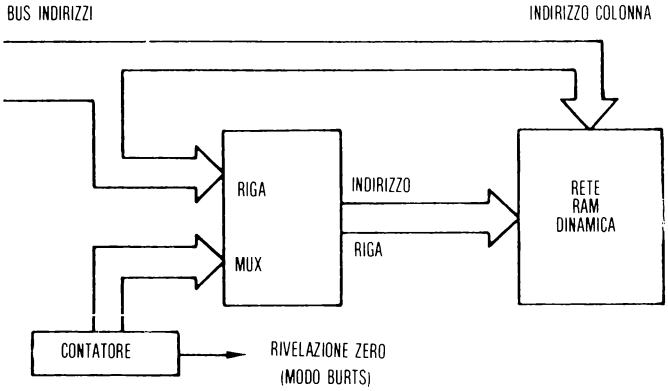


Fig. 4-146: Logica fondamentale per il ripristino

Contesa di memoria

Entrambe le tecniche richiedono di poter accedere alla memoria quando essa non é occupata e ad una priorità più alta del processore. Due tecniche fondamentali sono usate per realizzare tale sincronizzazione.

Accesso asincrono

Le richieste sono generate a velocità fissa, come ad esempio ogni 31 microsecondi (cioé 64 volte ogni 2 millisecondi) indipendentemente dallo stato del microprocessore. Questo metodo é indipendente dal tipo di microprocessore usato, ma richiede il progetto di una unità di controllo complessa e produce ritardi di accesso. Il controllore può dovere attendere il completamento di un ciclo RAM in corso; tuttavia il ritardo é nell'accesso alla memoria e non del microprocessore. Devono essere prese in considerazione priorità nelle ricerche e ritardi di propagazione attraverso la logica del controllore.

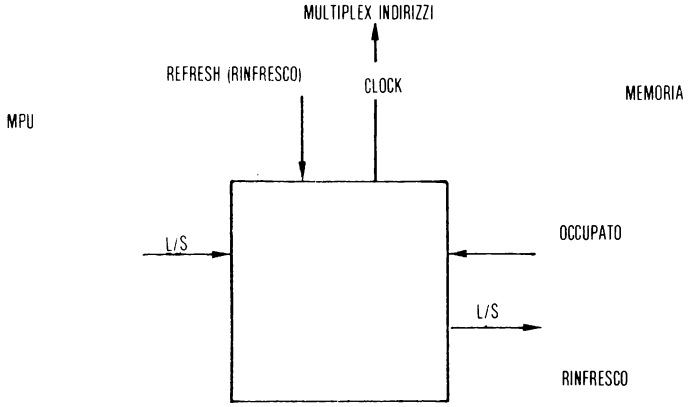


Fig. 4-147: Controllore asincrono

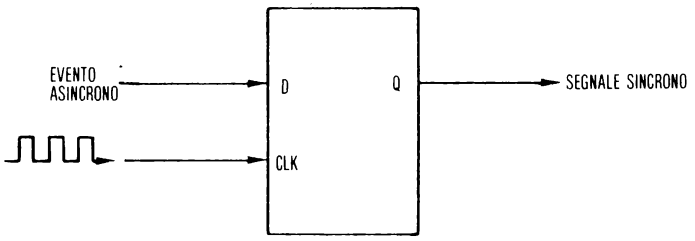


Fig. 4-148: Flip-flop usato come componente di sincronizzazione

Ripristino celato

Il principio di tale metodo consiste nel ripristinare la RAM quando il microprocessore non ha bisogno di essa. Il ripristino celato é anche conosciuto come *ripristino trasparente* o accesso sincrono. In tutti i microprocessori si riscontrano intervalli di uno o più cicli durante i quali non é richiesto accesso alla memoria. Se tali stati possono essere identificati all'esterno é possibile iniziare un ciclo di ripristino senza alcun problema di contese o ritardi di multiplazione e assenza di carico extra di gestione. Il microprocessore non viene a conoscenza del ripristino, e pertanto si usa l'aggettivo «celato» o «trasparente».

Il risultante vantaggio della velocità operativa é ovvio. Tuttavia il progetto della unità di controllo del ripristino dipende essenzialmente dalle caratteristiche del microprocessore utilizzato. Occorre inoltre considerare situazioni «inconsuete» che possono comportare il superamento dei due millisecondi consentiti per l'intervallo tra due ripristini. Eventi inconsueti sono, per esempio, nel caso dell'8080: Halt, Reset di durata eccessivamente lunga, lunghi Wait causati da memorie lente o da «debugging» a passo singolo, ed infine dallo stato di Hold dei cicli DMA.

Come esempio, lo stato T4 del ciclo di macchina M1 dell'8080A può essere usato per il ripristino: l'8080 decodifica ed esegue un'istruzione esterna durante T4. In tale intervallo non é pertanto usata la memoria. Può essere usato un semplice contatore per quattro per rivelare la fine di T3, e abilitare pertanto l'inizio del ripristino (vedi figura 4-150).

Inoltre, per qualsiasi microprocessore, l'autorizzazione al ripristino della RAM è garantita senza alcun accesso alla ROM. In figura 4-149 é indicato un esempio di un progetto sincrono, dove é usato il segnale HOLD per ottenere il controllo del bus. (Esso non può essere usato con RESET o WAIT).

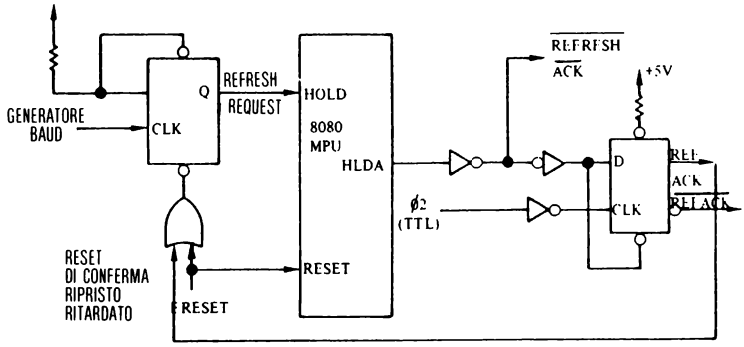


Fig. 4-149: Uso di HOLD per realizzare il furto di cicli di memoria

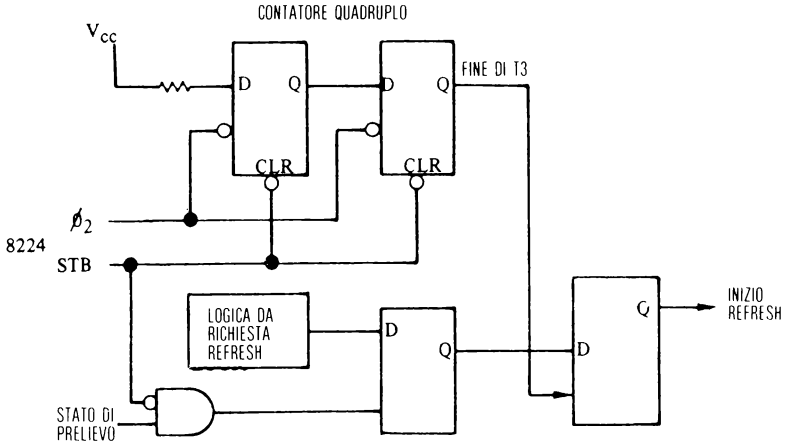


Fig. 4-150: Criterio di controllo del ripristino sincrono per l'8080

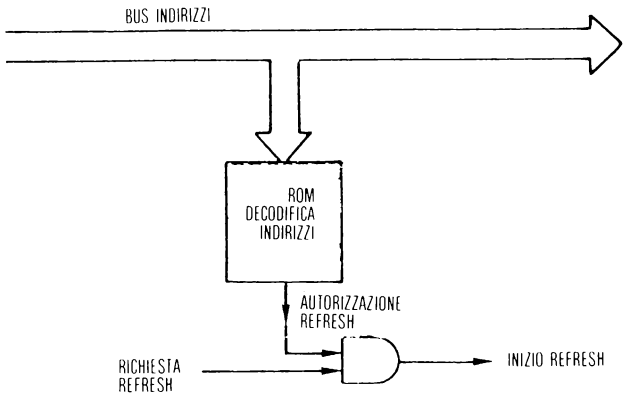


Fig. 4-151: Ripristino durante un accesso a ROM

Altri metodi

Possono essere usati un certo numero di altri metodi. Essi sono una combinazione di quelli già indicati, spesso traendo vantaggio dalle idiosincrasie del microprocessore specifico. Per esempio, un ripristino asincrono può essere temporizzato dalla transizione da livello basso ad alto di Ø2 (fase 2 del clock). Ciò garantisce nell'8080 che non ci sia contemporanea richiesta di accesso alla memoria da parte del microprocessore, e semplifica il progetto del circuito di abilitazione.

Logica del ripristino

Una unità di controllo del ripristino deve comprendere diversi elementi, secondo la efficienza richiesta e le limitazioni di costo. Gli elementi comuni sono:

- contatore del ripristino: un contatore a 6 o 7 bit, usato per generare sequenzialmente i 64 o 128 indirizzi di riga
- multiplatore di indirizzi: esso fornisce al modulo RAM l'indirizzo prodotto o dal contatore di ripristino (durante un ciclo di ripristino) o dal bus di indirizzi (durante un ciclo normale di memoria).
- unità di decisione sulla prima richiesta in ingresso: garantisce un ciclo di memoria alla unità di richiesta di ripristino o al microprocessore per una richiesta di accesso alla memoria (fig. 4-152).
- unità che tiene conto della priorità: garantisce sistematicamente accesso a memoria per una richiesta di ripristino, dipendente talvolta da specifiche condizioni.
- generatore di velocità in baud: circuito di temporizzazione che ha il compito di fornire impulsi alla velocità richiesta, cioé 64 volte ogni 2 millisecondi.
- controllori di trasferimento (latches): con il compito di memorizzare lo stato precedente.

Circuiti integrati di controllo del ripristino

Sono stati introdotti sul mercato circuiti integrati di controllo del ripristino per facilitare il progetto di controllori di RAM dinamiche. L'Intel 3222 é usato con il 2107B per realizzare una tecnica asincrona. Il 3222 richiede una temporizzazione esterna per i suoi segnali, ma comprende in un unico modulo controllori di trasferimento, l'oscillatore (richiede il circuito R-C esterno), il multiplatore di indirizzi, il contatore di ripristino, e il circuito che tiene conto delle priorità. Esso permette un indirizzo di riga a 6 bit (vedi fig. 4-153).

Il 3242 é un semplice controllore usato con il 2104A (4K) e il 2116 (16K). I bit forniti in uscita sono 6 o 7. Esso comprende inoltre il multiplatore di indirizzi e il contatore di ripristino (vedi fig. 4-154). Nel paragrafo successivo é descritto un progetto reale di una piastra di memoria dinamica per il bus S100.

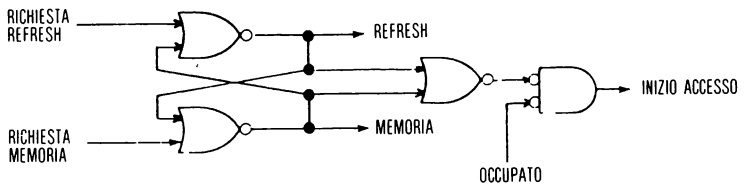


Fig. 4-152: Arbitro delle richieste

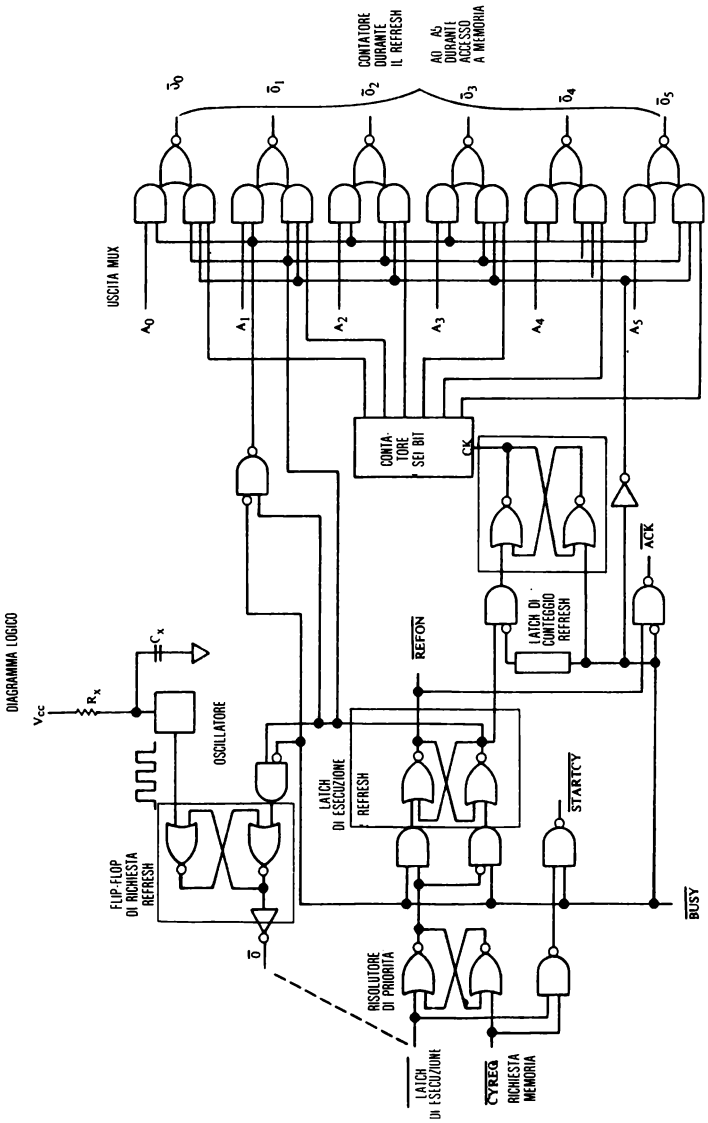


Fig. 4-153: Integrato di controllo del ripristino a 6 vie (Intel)

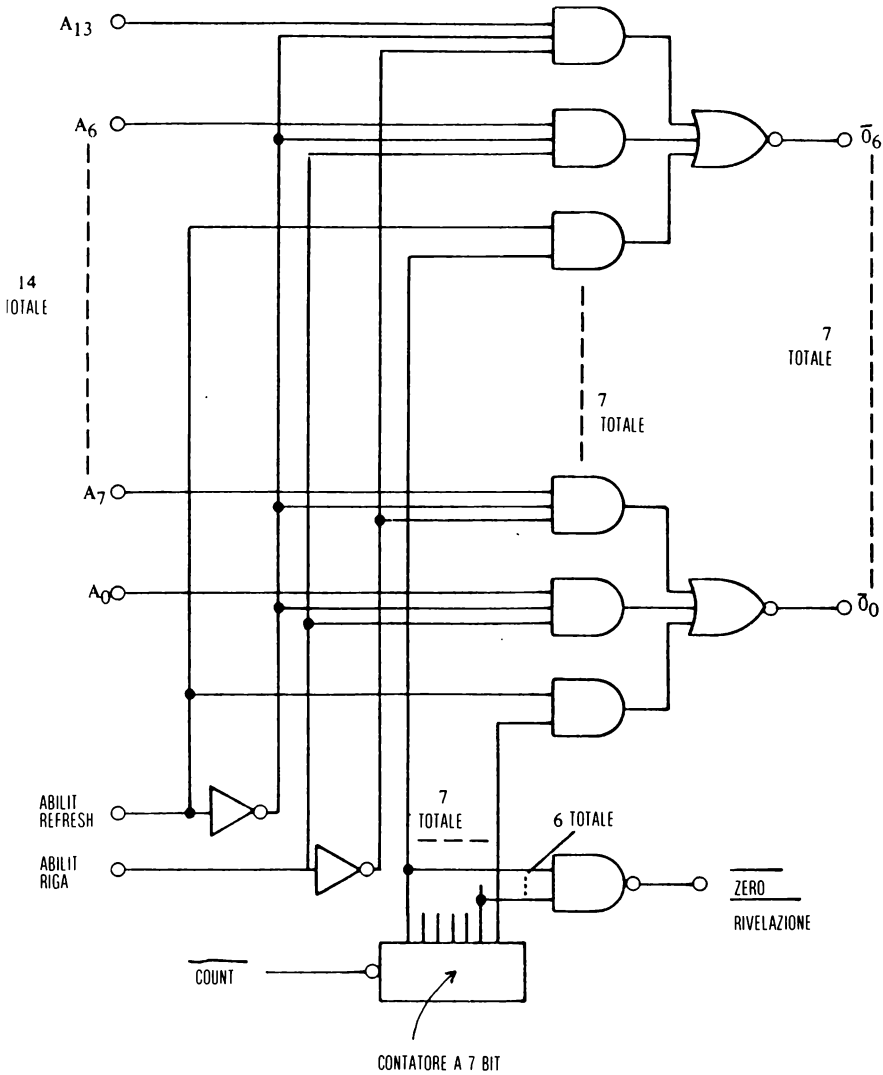


Fig. 4-154: Integrato di controllo del ripristino a 7 vie (Intel)

Riassumendo, possono essere usati tre metodi per il ripristino: *ripristino globale,* cioé tutte le trentadue righe ogni due millisecondi, *ripristino dedicato* per una colonna ad intervalli di alcuni microsecondi (consumando banda per accesso in memoria), e *ripristino trasparente* (ripristino celato) nel quale i cicli di ripristino sono eseguiti durante intervalli di istruzioni di sistema che non prevedono accesso alla memoria.

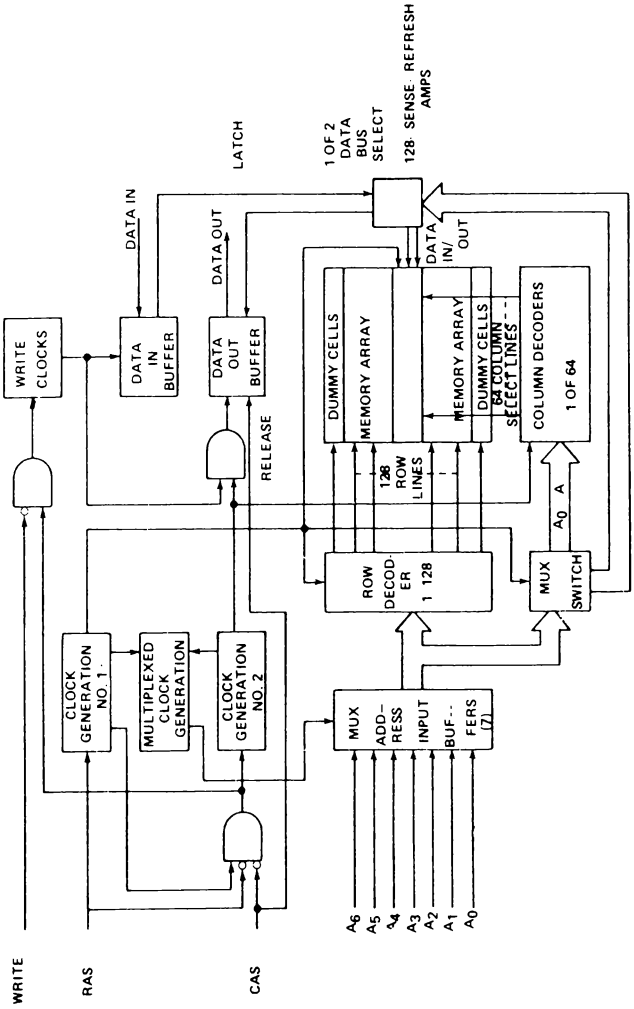


Fig. 4-155: Diagramma interno della RAM 4116

Progetto di una memoria dinamica per il bus S100

Il problema può essere diviso in due parti: la prima consiste nello studio delle specifiche del particolare modulo di memoria dinamica usato. La seconda consiste nella valutazione delle esigenze di interfaccia del bus S100 e di temporizzazione di memoria. Sfortunatamente devono essere considerati argomenti come i cicli in DMA, i cicli di accesso al pannello frontale e altri che non hanno nulla a che fare con il problema originale della temporizzazione del sistema 8080. I moduli di RAM dinamica usati nel caso in esame sono i Mostek 16.384 per un bit.

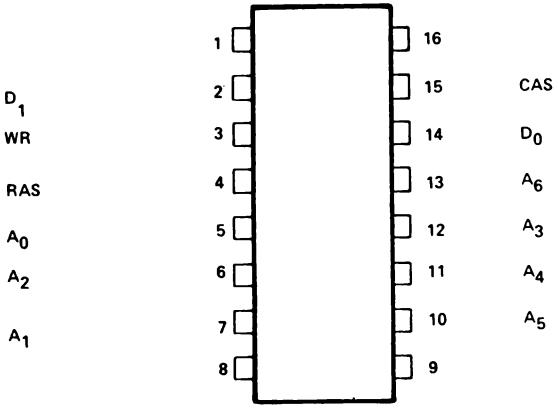


Fig. 4-156: Utilizzazione delle terminazioni del 4116

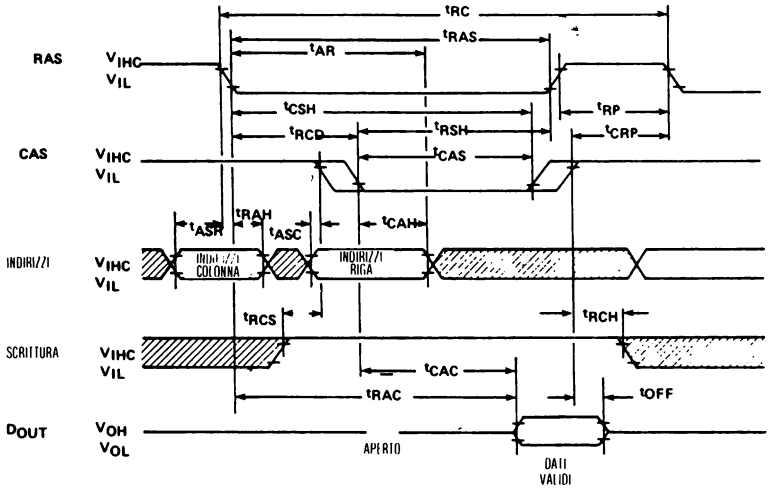


Fig. 4-157: Temporizzazione del ciclo di lettura

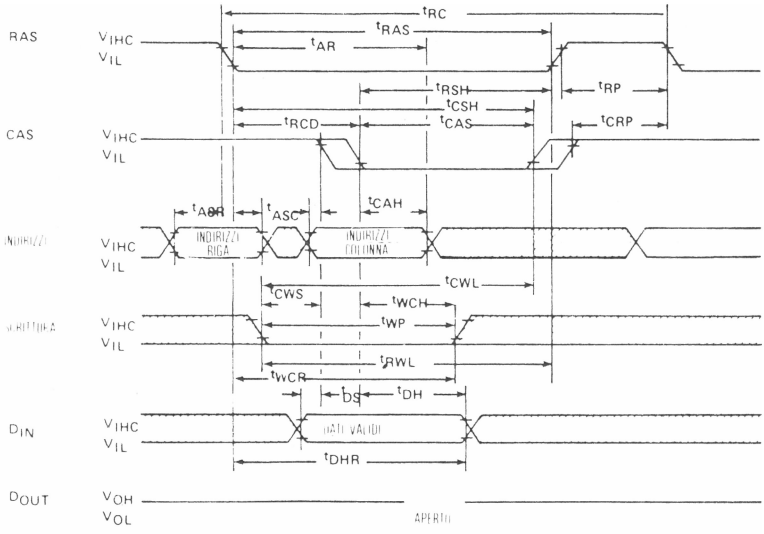


Fig. 4-158: Temporizzazione del ciclo di scrittura

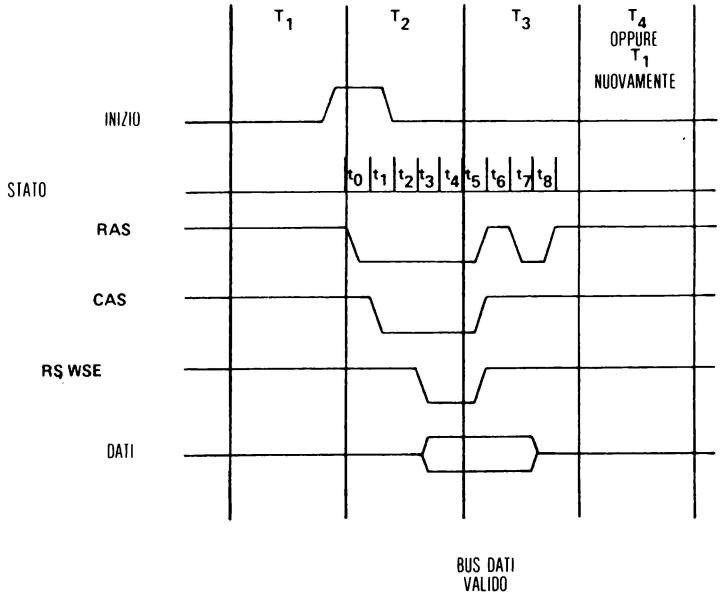


Fig. 4-159: Temporizzazione del controllore di stato

Il modulo RAM dinamica di 16.384 bit, lo MK4116P, é un integrato a 16 terminazioni. Poiché esso ha 16 terminazioni e sono necessari 14 bit per indirizzare la singola cella, gli indirizzi devono essere multiplati in due gruppi di sette. La figura 4-157 mostra il diagramma di temporizzazione per un ciclo di lettura nel sistema. La figura 4-158 mostra invece il diagramma di temporizzazione per un ciclo di scrittura nel sistema. La multiplazione degli indirizzi é realizzata attraverso i segnali RAS e CAS. L'unica informazione necessaria é relativa all'inizio di un ciclo di memoria di lettura o scrittura. Appena tale informazione é nota, é molto facile generare il segnale di temporizzazione mostrato in figura 4-159, usando un contatore di stato sincrono.

Il contatore di stato sincrono fornisce 8 impulsi di 100 nanosecondi uno di seguito all'altro. Ciascuno stato, da 0 a 7, é usato per generare i segnali di campionamento e controllo per le memorie dinamiche. Facendo riferimento al diagramma schematico del sistema indicato in figura 4-160, sono presenti 8 moduli di memoria, 4 controllori di trasferimento in uscita (latch), una memoria tampone per i dati in ingresso, la multiplazione degli indirizzi e il contatore di ripristino. Tutti i segnali di controllo indicati sono generati dall'unità di controllo di temporizzazione in fig. 4-161. La figura 4-161 ha il contatore di stato che é fatto scattare da ogni ciclo di memoria di scrittura o lettura.

Il progetto procede in modo semplice. Il contatore di stato sincrono fornisce al progettista flessibilità nel controllo della temporizzazione senza i problemi presenti

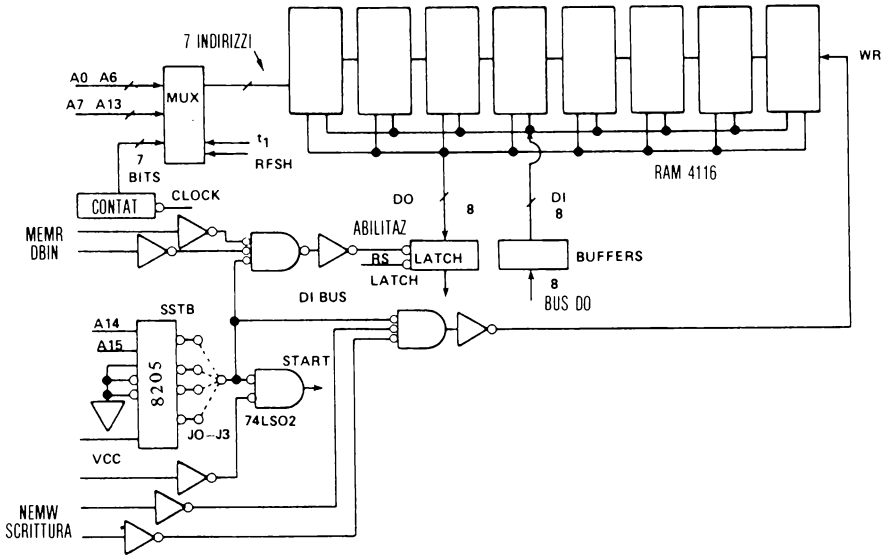


Fig. 4-160: Schema di una piastra, di memoria di 16K

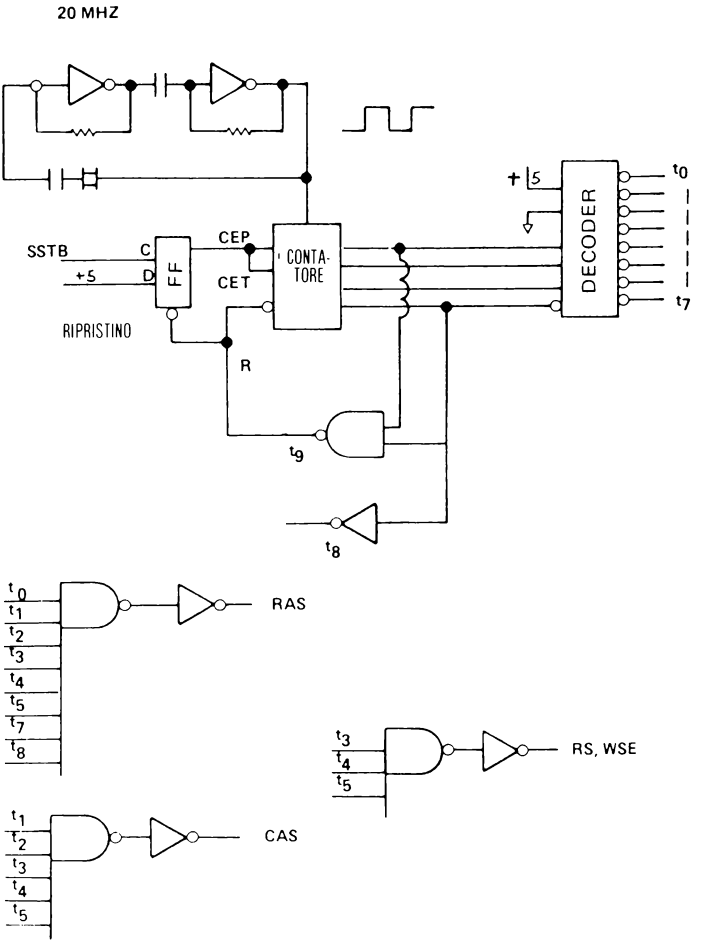


Fig. 4-161: Sezione di controllo della piastra di memoria di 16K

se si fa uso di componenti con temporizzazione asincrona. Deve essere definito il numero opportuno di cicli necessari per generare sia i segnali RAS e CAS che altri necessari.

Come esempio, figura 4-159, il segnale RAS é «falso» durante t0, t1, t2, t3, t4, t5 e t7, t8; il segnale CAS é invece falso durante t1, t2, t3, t4, t5. È da osservare che il segnale RAS é falso durante t0 mentre CAS é vero durante lo stesso intervallo. In base al diagramma di figura 4-159 ciò permetterà di multiplare gli indirizzi inizialmente per le righe, poi per le colonne, entro il registro degli indirizzi interno al circuito integrato. Pertanto il segnale di temporizzazione t0 diventa la finestra di presentazione al multiplatore degli indirizzi del contenuto sul bus corrispondente.

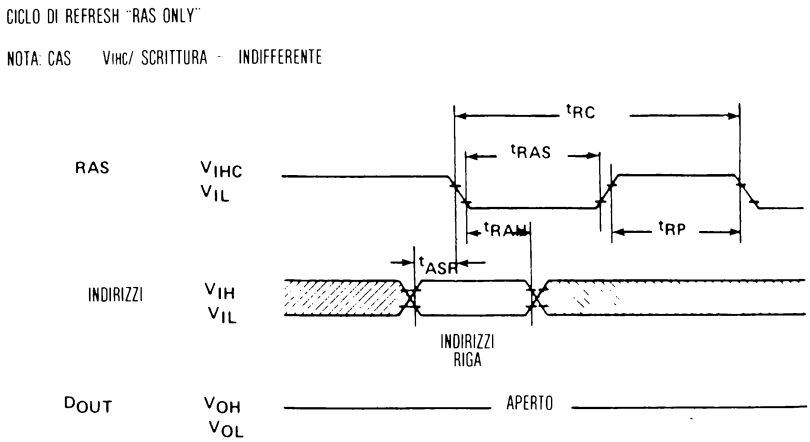


Fig. 4-162: Temporizzazione del ripristino.

Adesso occorre scegliere il metodo di ripristino. In tal caso, naturalmente sarebbe preferibile il ripristino di tipo celato, poiché esso resterebbe «celato» al microprocessore. Per ottenere un ripristino trasparente la memoria è ripristinata ad ogni ciclo di lettura e di scrittura. Rimandando alle temporizzazioni dei cicli di memoria di scrittura e di lettura sviluppate nel capitolo sei, figure 6-9 e 6-10, nel peggiore dei casi una lettura o una scrittura deve essere realizzata entro 650 nanosecondi e il ripristino deve essere completo prima dell'inizio del ciclo M-esimo successivo. Facendo riferimento alla figura 4-159 sono disponibili 400 nanosecondi per realizzare un ripristino celato, in assenza di stati di attesa (Wait). Queste particolari memorie dinamiche sono sufficientemente veloci da permettere il completamento del ciclo celato entro i 400 nanosecondi. Per far questo, dopo che é stato completato ogni ciclo di lettura o scrittura, é incrementato il contatore di ripristino ed é attivato il segnale RAS per 200 nanosecondi per realizzare il ripristino di una riga. Nel caso peggiore restano soltanto 50 nanosecondi prima di un nuovo potenziale ciclo di memoria.

Restano pertanto soddisfatte le esigenze dei cicli base di scrittura e lettura dell'S100. È stato usato il segnale di presentazione campionata dello stato. Questo unico segnale è la chiave dell'intero sistema di ripristino. Se è usato un qualsiasi altro sistema diverso dall'8080 il campionamento indicato deve soddisfare le esigenze di temporizzazione necessarie per il sistema base 8080. Oltre a ciò, ogni altra operazione, come quella su pannello frontale, o cicli di DMA, devono anche utilizzare la stessa sequenza di stato. Sfortunatamente, non esistono accordi tra i costruttori del bus S100 sulle reali esigenze di cicli di scrittura e di lettura in tali circostanze alternative. A causa di tali incompatibilità di temporizzazione, alcune piastre di me-

PARAMETER	SYMBOL	MK 4116-2 MIN	MAX	UNITS
RANDOM READ OR WRITE CYCLE TIME	t_{RC}	375		ns
READ–WRITE CYCLE TIME	t_{RWC}	375		ns
PAGE MODE CYCLE TIME	t_{PC}	170		ns
ACCESS TIME FROM $\overline{RAS}$	t_{RAC}		150	ns
ACCESS TIME FROM $\overline{CAS}$	t_{CAC}		100	ns
OUTPUT BUFFER TURN -OFF DELAY	t_{OFF}	0	40	ns
TRANSITION TIME (RISE AND FALL)	t_T	3	35	ns
$\overline{RAS}$ PRECHARGE TIME	t_{RP}	100		ns
$\overline{RAS}$ PULSE WIDTH	t_{RAS}	150	10,000	ns
$\overline{RAS}$ HOLD TIME	t_{RSH}	100		ns
$\overline{CAS}$ HOLD TIME	t_{CSH}	150		
$\overline{CAS}$ PULSE WIDTH	t_{CAS}	100	10,000	ns
$\overline{RAS}$ TO $\overline{CAS}$ DELAY TIME	t_{RCD}	20	50	ns
$\overline{CAS}$ TO $\overline{RAS}$ PRECHARGE TIME	t_{CRP}	- 20		ns
ROW ADDRESS SET--UP TIME	t_{ASR}	0		ns
ROW ADDRESS HOLD TIME	t_{RAH}	20		ns
COLUMN ADDRESS SET-UP TIME	t_{ASC}	−10		ns
COLUMN ADDRESS HOLD TIME	t_{CAH}	45		ns
COLUMN ADDRESS HOLD TIME REFERENCED TO $\overline{RAS}$	t_{AR}	95		ns
READ COMMAND SET UP TIME	t_{RCS}	0		ns
READ COMMAND HOLD TIME	t_{RCH}	0		ns
WRITE COMMAND HOLD TIME	t_{WCH}	45		ns
WRITE COMMAND HOLD TIME REFERENCED TO $\overline{RAS}$	t_{WCR}	95		ns
WRITE COMMAND PULSE WIDTH	t_{WP}	45		ns
WRITE COMMAND TO $\overline{RAS}$ LEAD TIME	t_{RWL}	60		ns
WRITE COMMAND TO $\overline{CAS}$ LEAD TIME	t_{CWL}	60		ns
DATA-IN SET--UP TIME	t_{DS}	0		ns
DATA-IN HOLD TIME	t_{DH}	45		ns
DATA-IN HOLD TIME REFERENCED TO $\overline{RAS}$	t_{DHR}	95		ns
$\overline{CAS}$ PRECHARGE TIME (FOR PAGE--MODE CYCLE ONLY)	t_{CP}	60		ns
REFRESH PERIOD	t_{REF}		2	ms
$\overline{WRITE}$ COMMAND SET-UP TIME	t_{WCS}	−20		ns
$\overline{CAS}$ TO $\overline{WRITE}$ DELAY	t_{CWD}	70		ns
$\overline{RAS}$ TO $\overline{WRITE}$ DELAY	t_{RWD}	120		ns

Fig. 4-163: Definizioni di temporizzazione della RAM 4116

moria dinamica non possono essere usate in tutti i sistemi. Esempio di una piastra che usa lo stesso metodo di temporizzazione del ripristino celato mediante uno stato sincrono è la Dynabyte con 16K di memoria. Questa piastra usa un modulo RAM dinamico a 4K. Essa ha la capacità di regolare la temporizzazione in modo da poter essere interfacciata a molte altre applicazioni del bus S100, inclusi processi di diverso tipo. In figura 4-164 è indicata una fotografia della piastra. In figura 4-165 è invece indicato un diagramma a blocchi dello schema di controllo di questa piastra. È da notare che sono usati molti segnali di stato provenienti dal bus per determinare l'attuale ciclo operante. Ciò significa che ogni segnale DMA deve anche produrre tali segnali.

In conclusione, è stato realizzato un sistema a memoria dinamica badando alle esigenze di temporizzazione del modulo e a quelle del sistema. L'uso di un sistema di stato sincrono semplifica e definisce le temporizzazioni del sistema. Come esempio, si è qui assunto che il progettista ha accesso ai segnali di temporizzazione del

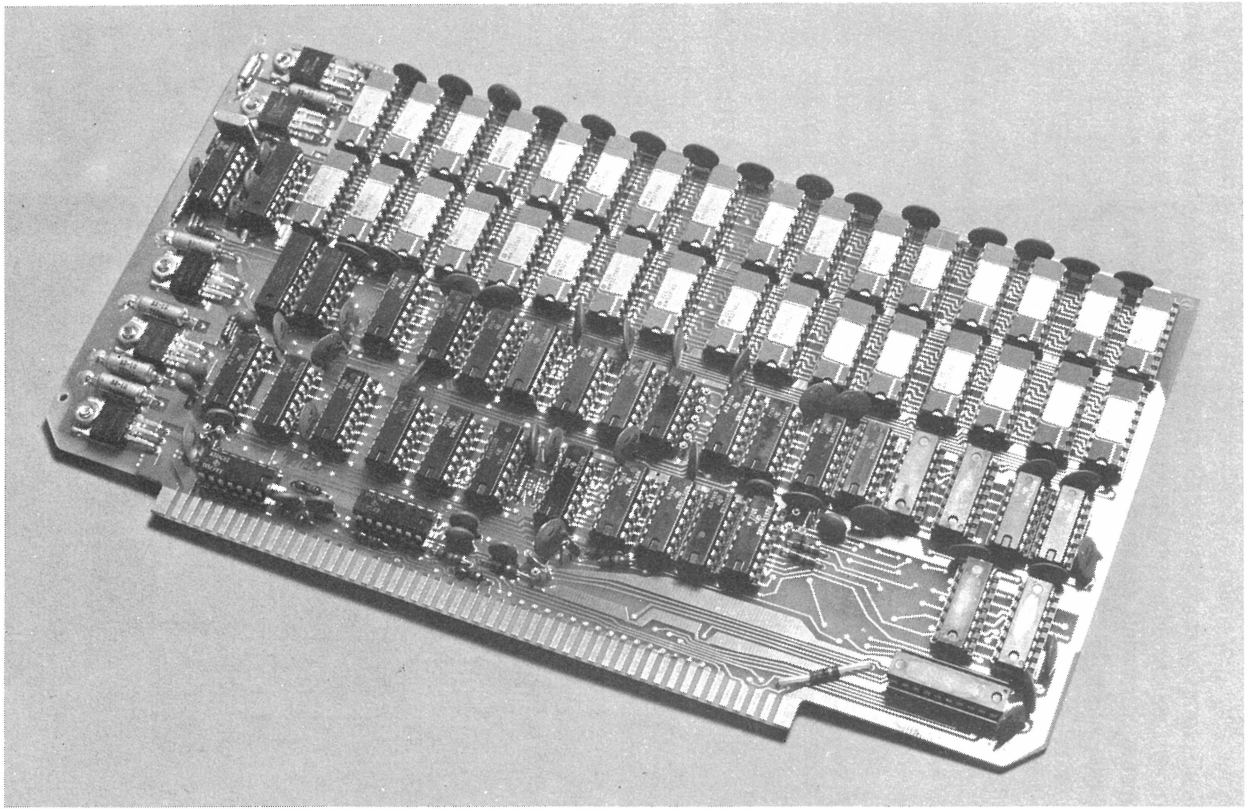


Fig. 4-164: Piastra RAM di 16K della Dynabite

sistema, in modo da poter conoscere in tutte le condizioni cosa il sistema sta facendo. I progetti potrebbero usare monostabili asincroni, celle di ritardo R-C o linee di ritardo; tuttavia tali progetti sarebbero sensibili a problemi di temporizzazione per la tolleranza dei componenti, a cambiamenti di temperatura, etc.

L'indicata complicazione dovuta alla presenza di stati WAIT, di differenti sistemi di temporizzazione e l'assenza di alcuni segnali di stato sul bus tra l'altro non universalmente concordati, fanno sì che non esistano piastre di RAM dinamica di uso generale su un Bus S100.

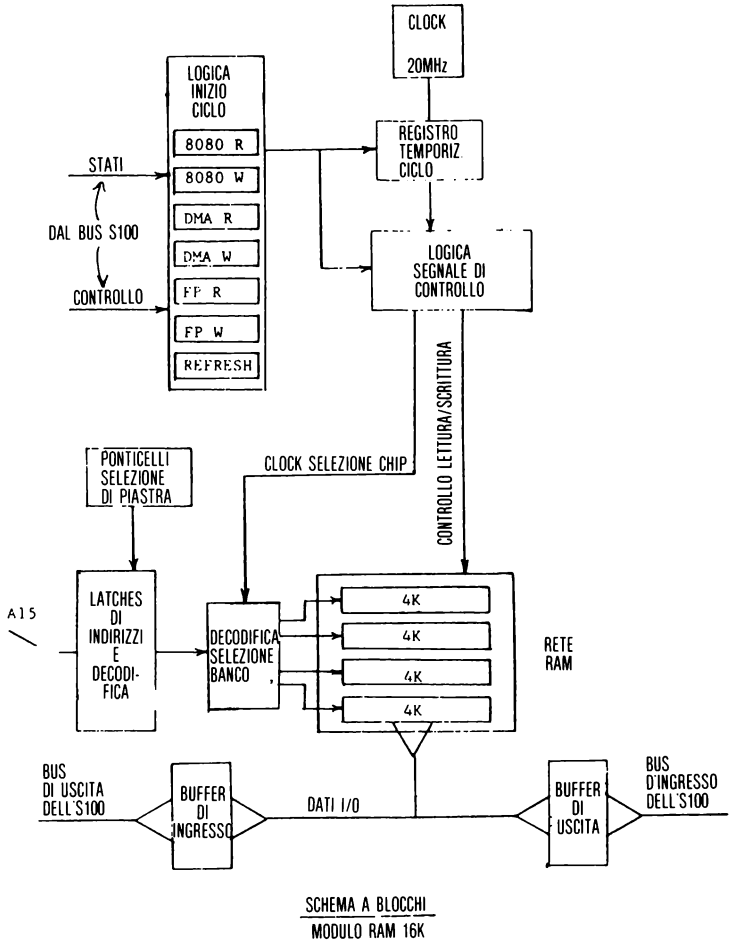


Fig. 4-165: Diagramma a blocchi della piastra Dynabyte

La piastra più interessante, quella prodotta dalla Dynabyte, può essere configurata in modo da poter lavorare in particolari sistemi mediante l'uso di ponticelli hardware. Ci sono tuttavia alcuni sistemi per i quali il livello di flessibilità realizzato non è sufficiente.

In contrasto, se si guarda al bus 6800 per l'Altair 680B, si osserva che il sistema di temporizzazione è così ben definito da semplificare notevolmente i problemi di interfacciamento.

Il progetto di una piastra di memoria dinamica è spesso reso ancora più complesso dalla necessità di tre livelli di alimentazione e dalla presenza di rumore. Infatti un tale progetto spesso non ha buon esito per l'impossibilità di leggere o scrivere informazioni nella memoria. Pertanto il progetto del sistema di memoria dinamica deve essere curato attentamente sotto i diversi aspetti indicati, includendo il problema complesso dei picchi di rumore ad alta frequenza prodotti dal normale funzionamento del circuito integrato. Il miglior riferimento per il progetto di un sistema di memoria dinamica è il catalogo delle memorie prodotto dal costruttore. Poichè le memorie sono il pane dei produttori di semiconduttori, essi sono più che felici di aiutare i progettisti di piastre di memoria nell'uso dei loro prodotti.

CONCLUSIONI

L'evoluzione degli UART, i PIA e altri moduli LSI, verso FDC e CRTC traccia le tappe dello sviluppo tecnologico. Un numero maggiore di controllori di periferiche saranno totalmente integrati in un singolo modulo, e un numero maggiore di controllori saranno «intelligenti». Diventeranno caratteristiche standard delle periferiche del futuro la composizione locale della stampa, la disponibilità di librerie di file, e l'elaborazione dei testi.

Fig. 5-0: Piastre di convertitori A/D, D/A

CAPITOLO 5

CONVERSIONE ANALOGICA-DIGITALE E DIGITALE-ANALOGICA

INTRODUZIONE

In ogni sistema devono essere generati o misurati due tipi fondamentali di segnali. Essi sono i segnali *analogici* e *digitali.*

I segnali *analogici* assumono una gamma *continua* di valori, mentre i segnali *digitali* assumono solo un *numero limitato* di valori. Come esempio un segnale binario è un segnale digitale il quale assume uno dei due valori, «on» o «off» «1» o «0». Un tipico esempio di un segnale analogico é il valore della temperatura di un forno. La temperatura, essendo una variabile analogica, può assumere un infinito numero di valori intermedi.

A causa della limitata memoria di un calcolatore elettronico e della sua precisione finita, sarà utilizzata una rappresentazione digitale.

Normalmente si dice che la precisione della misura è limitata ad N digits significativi. In aggiunta si userà la *campionatura* per ridurre la memoria richiesta. Il concetto di campionatura sarà discusso più avanti.

Questo capitolo spiegherà come effettuare le conversioni analogico - digitali (A/D) e digitali-analogiche (D/A). Inoltre saranno spiegati gli specifici componenti richiesti per costruire un sistema per la raccolta di dati. Considerando nell'ordine:

- un convertitore D/A (o DAC).
- un convertitore A/D (o ADC).
- un processo di campionamento.
- un multiplatore analogico.

Infine tutte queste tecniche saranno usate per costruire un sistema completo per la raccolta di dati.

CONCETTO DI D/A

Consideriamo il problema di convertire un numero binario in una tensione analogica. Questo è un problema tipico della conversione digitale analogica.

Una soluzione semplice è la seguente: si genera una tensione per ogni posizione di bit all'interno del numero binario. Il valore della tensione è proporzionale al peso binario del bit.

Ad esempio, il bit 0 genererà una tensione V (2^0); il bit 1 genererà una tensione 2 V (2^1); il bit 2 genererà una tensione 4 V (2^2); ed il bit n genererà una tensione 2^n xV. Le tensioni che ne risultano sono sommate direttamente. Il risultato é proporzionale al numero binario originale.

Un semplice convertitore D/A a quattro bit è mostrato in fig. 5-1. Questo convertitore D/A consiste di: quattro interruttori, quattro resistenze, un amplificatore operazionale ed una resistenza di controreazione.

I valori delle resistenze sono nella proporzione 1, 2, 4, 8. Ne risultano guadagni di: -1/8, -1/4, -1/2 e -1. Esaminiamo le funzioni del circuito.

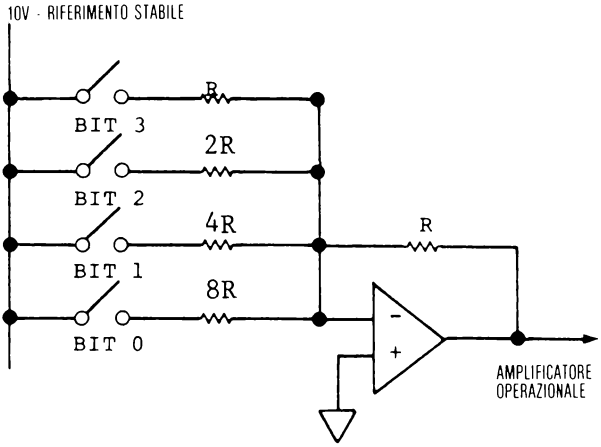


Fig. 5-1: Un semplice convertitore D/A a 4 bit

Iniziamo con tutti gli interruttori nella posizione di *aperto*. Poiché non vi è ingresso nell'amplificatore operazionale l'uscita sarà «0». Chiudendo l'interruttore del bit numerato «0» applicheremo la tensione di −10 V all'ingresso dell'amplificatore operazionale, attraverso la resistenza indicata con 8R. Ne risulterà una tensione all'uscita di 1,25 V (dovuta al guadagno di −1/8). Chiudendo l'interruttore indicato con «bit 1» sommeremo 2,5V al precedente valore di 1,25V (dovuto al guadagno di -1/5). La tensione di uscita è 3,75V.

Se tutti gli interruttori sono *chiusi* la conseguente tensione di uscita è 10,0 + 5,0 + 2,5 + 1,25 ossia 18,75 volt. Così abbiamo convertito un numero binario di 4-bit, rappresentato mediante i quattro interruttori di una tensione. Ciò è la rappresentazione analogica di uno dei 16 possibili valori digitali.

Esaminiamo adesso la struttura di un convertitore D/A reale.

Convertitori D/A usati in pratica

Lo schema di fig. 5-2 mostra il tipico schema di un convertitore D/A monolitico. Questo dispositivo ha quattro bit di risoluzione. In pratica si sommano correnti al posto di tensioni perché le correnti sono più facili da attivare e da interrompere con precisione. Per ottenere all'uscita una tensione si pone all'ultimo stadio del convertitore un *convertitore corrente-tensione.*

Questo si ottiene facilmente usando un amplificatore operazionale. I convertitori tipici hanno otto bit di risoluzione.

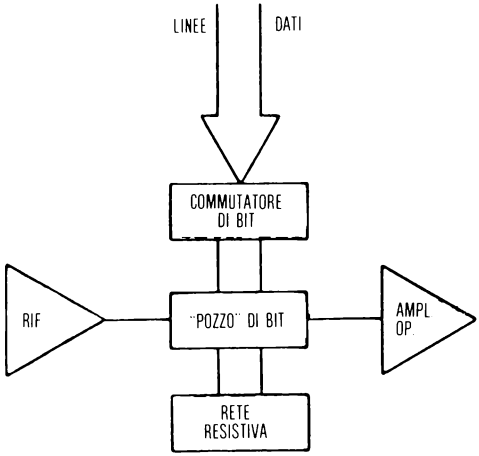


Fig. 5-2: Diagramma di un convertitore reale

La figura 5-3 mostra gli elementi funzionali del nostro convertitore. Essi sono la sorgente della corrente di riferimento, i transistor a derivazione di bit, l'insieme dei resistori a valori scalati, gli interruttori a derivazione di bit, ed il convertitore tensione-corrente. La corrente di riferimento a derivazione di bit costituisce una stabile corrente di riferimento.

Le sorgenti della corrente a derivazione di bit saranno proporzionali a questa corrente di riferimento. La corrente in ogni transistor a derivazione di bit è determinata dalla sua posizione nell'insieme a scala che va da R a 2R. L'insieme delle resistenze a scala genera una serie di correnti 2^{-n} che circola attraverso ogni collettore di derivazione del bit. Gli interruttori invieranno la corrente al bus di derivazione del bit che collega il convertitore.

Nel nostro esempio queste correnti sono 1/20 ampere, 1/40 ampere, 1/80 ampe-

re, 1/160 ampere. Questi elementi combinati fra loro contribuiscono ad operare la conversione da un numero binario a 4-bit ad un voltaggio analogico.

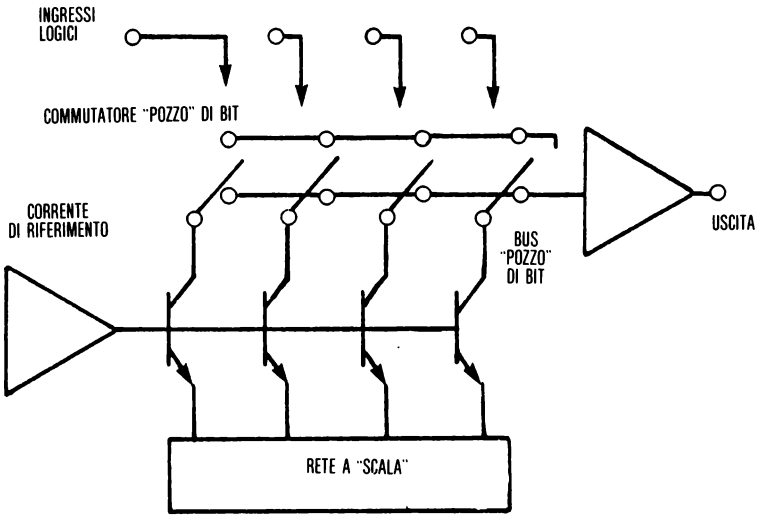


Fig. 5-3: Elementi funzionali di un convertitore monolitico

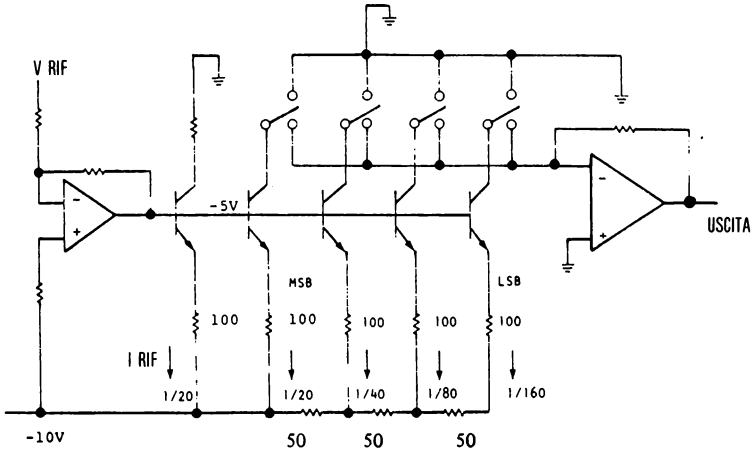


Fig. 5-4: Convertitore monolitico completo

Un convertitore monolitico usa dei transistor come interruttori per inviare la corrente dal bus di derivazione del bit all'amplificatore ed alla massa.

La fig. 5-4 mostra la circuiteria di interfaccia tra il segnale logico e gli interruttori di derivazione di bit di corrente.

Quando l'ingresso è uno «0» logico, che corrisponde a 0 volt, la derivazione di bit preleverà la corrente dal bus di derivazione del bit attraverso Q4. Quando l'ingresso è «1» logico, che corrisponde ad una tensione di ingresso maggiore di 2 volt la derivazione di bit preleverà corrente attraverso Q3 invece di Q4 interrompendo il bus di derivazione di bit da questa derivazione stessa.

I quattro segnali binari inseriscono e disinseriscono le quattro derivazioni di bit dal bus. La corrente risultante è convertita in tensione di uscita.

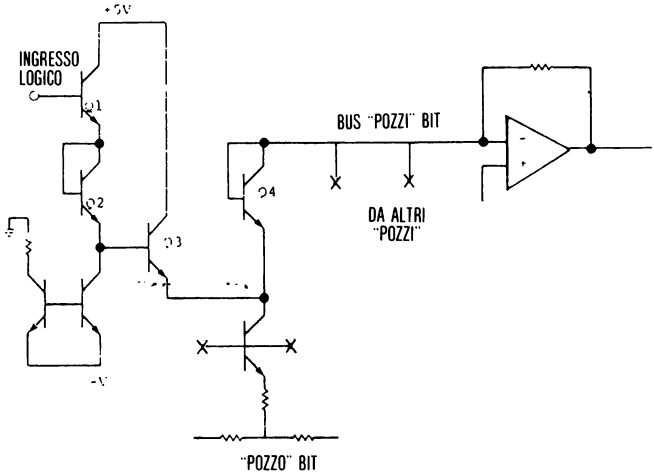


Fig. 5-5: Dettaglio: commutatori di bit

Estendendo l'insieme delle resistenze a scala R-2R ed aggiungendo più di un transistor di derivazione di bit, si può aumentare la risoluzione del nostro convertitore a più di 10 bit.

Con più di 14 bit vi sono problemi di stabilità che questo semplice circuito non può risolvere. Infatti i convertitori a 16 bit generalmente sono garantiti in linea con uno standard nazionale (si deve ricordare che 16 bit danno una esattezza di 1 su 65.000!).

Prodotti in commercio

La figura 5-6 rappresenta un elenco di alcuni prodotti che operano la conversione D/A. Il loro costo cresce con la velocità.

COSTRUTTORE	TIPO#	RISOLUZIONE	VELOCITÀ
Motorola	MC1408	8	300ns
PMI	DAC-08	8	100ns
PMI	DAC-03	10	250ns
Analog Devices	AD7520	10	500ns
Datel	DAC-4Z12D	12	1us
Burr-Brown	DAC70/CSB	16	75us

Fig. 5-6: Convertitori D/A

CONVERSIONE A/D

Ora che abbiamo convertito la rappresentazione binaria di un numero in un segnale analogico dobbiamo risolvere il problema inverso.

Dobbiamo misurare un segnale analogico e convertirlo in un segnale binario. Ci sono tre modi di conversione: per approssimazioni successive, per integrazione e per confronto diretto. Prima di studiare questi metodi esaminiamo il concetto di campionamento.

Campionamento

Il numero binario che rappresenta il nostro segnale analogico, rappresenta un valore ad un certo istante. Questo é detto *campione*. Nella seguente forma d'onda di fig. 5-7 si è campionato il segnale dove indicato dalle frecce. I valori dei campioni non danno nessuna informazione sulla vera forma d'onda del segnale analogico.

Si devono raccogliere campioni con i quali si può rappresentare accuratamente il segnale analogico. La frequenza con la quale si raccolgono campioni è detta *velocità di campionamento*. Per rappresentazioni più accurate del segnale dobbiamo campionare con frequenza più elevata. Qual'è la frequenza di campionamento?

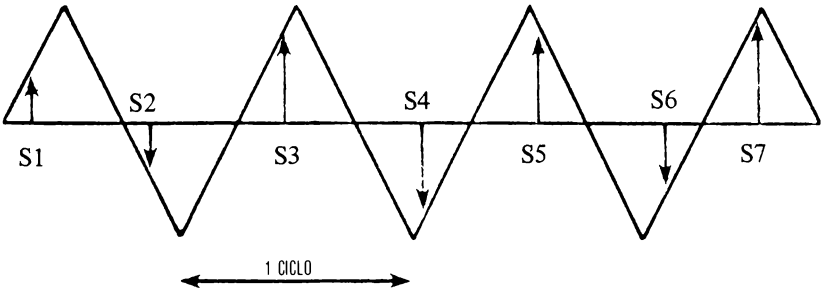


Fig. 5-7: Campionamento poco frequente

Teorema del campionamento

La risposta alla domanda precedente è data dal teorema del campionamento: *si deve campionare al minimo con frequenza doppia del segnale a frequenza più elevata che interviene nel sistema.* Una regola pratica dice di campionare almeno 10 volte più velocemente della frequenza media. La fig. 5-8 illustra l'effetto di un campionamento più veloce.

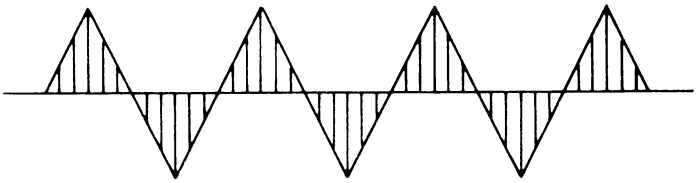


Fig. 5-8: Campionamento frequente

Campionamento e mantenimento dei segnali

L'ingresso analogico di un convertitore deve essere mantenuto stabile per il tempo di conversione. Questo può essere fatto usando un circuito di campionamento e mantenimento. Questo circuito campiona l'ingresso analogico e lo mantiene costante fino al prossimo campionamento. Il circuito conserva il segnale in un condensatore di alta qualità, tamponato da un amplificatore operazionale. I circuiti di campionamento e di conservazione sono ambedue disponibili sia in forma monolitica che ibrida.

CONVERSIONE A/D MEDIANTE APPROSSIMAZIONE SUCCESSIVA

Usando un circuito D/A si può eseguire una conversione A/D. Si può comparare il segnale analogico sconosciuto con un segnale *«ipotizzato»*. Variando il segnale ipotizzato, basandoci sulla conoscenza che il segnale incognito è più grande o più piccolo, si può convergere verso la conoscenza del corretto valore del segnale analogico. In fig. 5-9 si è connessa l'uscita del convertitore D/A a 4-bit all'ingresso di un *comparatore*. L'altro ingresso è connesso al segnale analogico sconosciuto. L'uscita del comparatore sarà «0» se il segnale sconosciuto è minore dell'uscita del convertitore D/A, oppure sara «1» se il segnale sconosciuto è maggiore della uscita del convertitore D/A.

L'algoritmo di «ipotizzazione» è quello di alzare un bit alla volta in maniera successiva all'interno del nostro numero binario partendo dal bit più significativo (MSB). Come alziamo un bit si verifica se siamo maggiori o minori guardando l'uscita del comparatore. Se siamo minori si lascia alzato il bit, se siamo maggiori lo si

abbassa. In ogni caso si passa al prossimo bit meno significativo fino che non arriviamo all'ultimo bit.

La fig. 5-10 illustra una tale procedura di comparazione con un convertitore a 4-bit. La fig. 5-11 illustra il diagramma di flusso per la stessa procedura di comparazione. Questa procedura è conosciuta come «conversione analogico-digitale per successiva approssimazione».

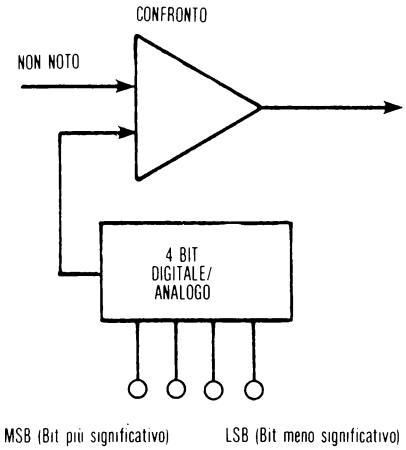


Fig. 5-9: Hardware per approssimazioni successive

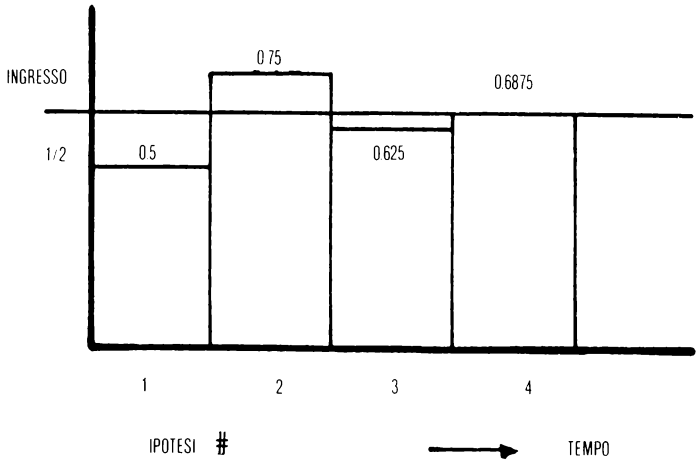


Fig. 5-10: Sequenza temporale di valori supposti: approssimazione successiva

Usando questo metodo si devono fare tante ipotesi quanti sono i bit del numero binario da convertire. Questo è il metodo più comune per conversioni A/D.

Ci sono due vie per eseguire l'approssimazione successiva: usando l'hardware e usando software più hardware. La fig. 5-12 illustra uno schema A/D che usa un *Registro di approssimazioni successive* o SAR. Questo registro esegue lo sposta-

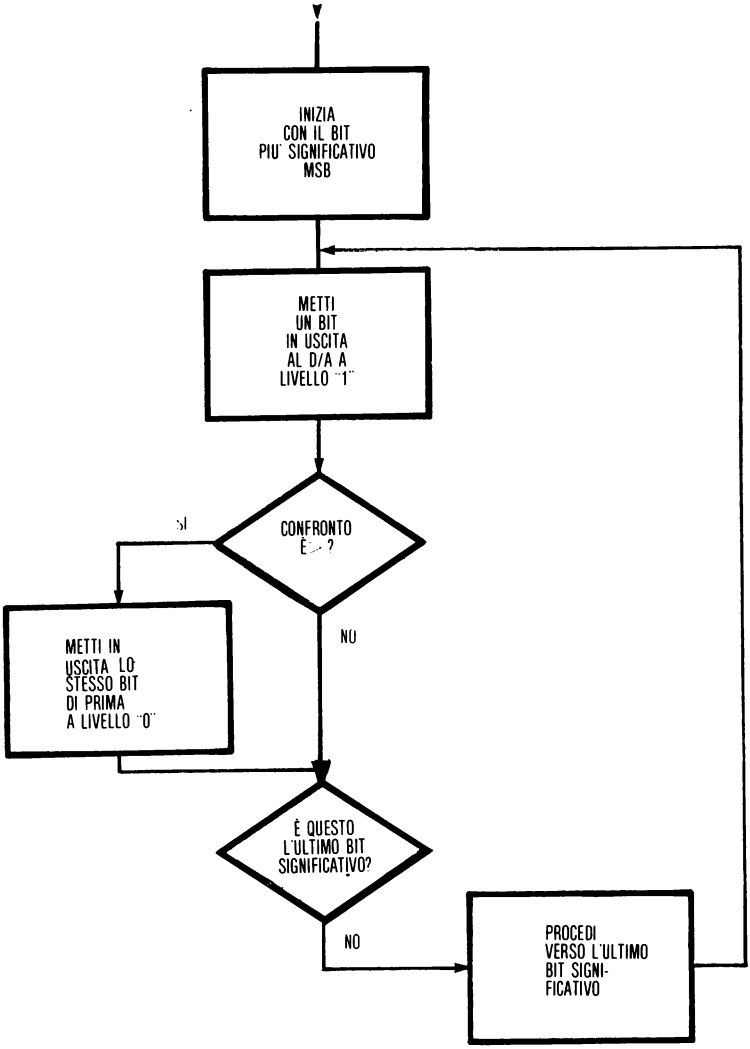


Fig. 5-11: Diagramma di flusso di approssimazione successiva

mento dei Bit (Bit-Shift) via hardware. L'altra alternativa è di far fare la funzione di approssimazione successiva A/D ad un algoritmo *software.*

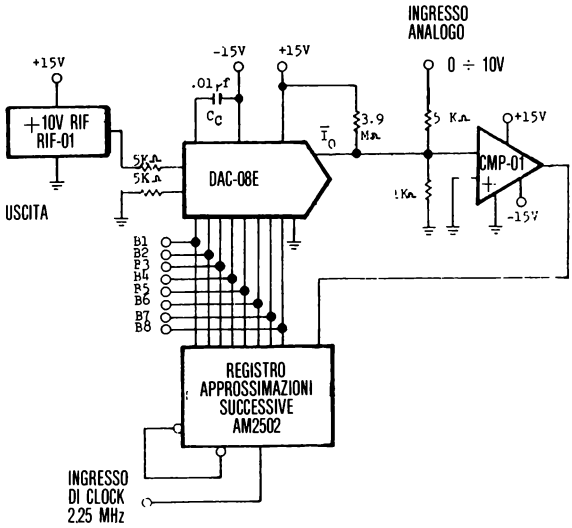


Fig. 5-12: Convertitore A/D che fa uso di un SAR

Un convertitore A/D monolitico senza il riferimento, ma con un «registro di approssimazione successiva» è illustrato in fig. 5-13.

Oltre alla tecnica di approssimazione successiva per la conversione si usano due altre tecniche: integrazione e comparazione diretta.

INTEGRAZIONE

Il secondo metodo di eseguire la conversione analogico-digitale è *l'integrazione analogica.* Questo metodo si basa sulla misura del tempo necessario ad un condensatore per caricarsi ad una tensione incognita e del tempo per scaricarsi fino ad una tensione di riferimento. Il rapporto fra la tensione incognita e quella nota è uguale al rapporto dei due tempi sopra misurati.

In pratica si fa l'integrale di una tensione positiva sconosciuta con un riferimento negativo noto. La tensione positiva risulterà incrementare la carica del condensatore. Dopo un periodo di tempo noto si applicherà all'integratore la tensione di riferimento e sarà misurato il tempo richiesto perché la carica del condensatore vada a «0».

La fig. 5-14 illustra il diagramma di temporizzazione di questa tecnica.

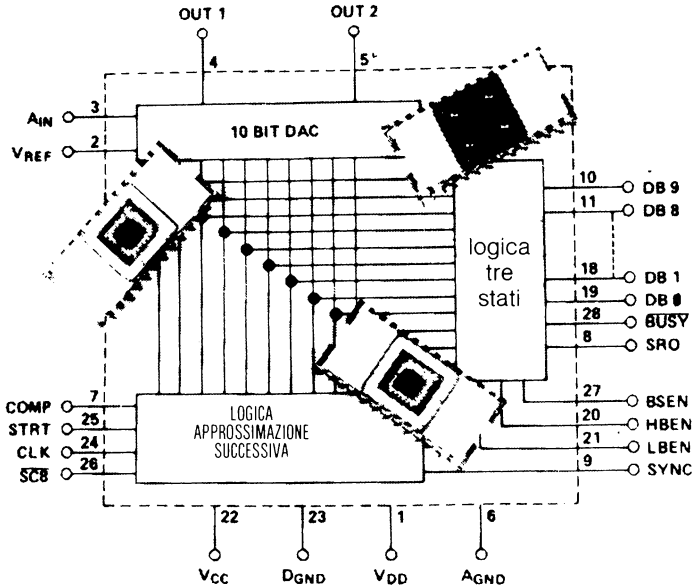


Fig. 5-13: Convertitore monolitico A/D

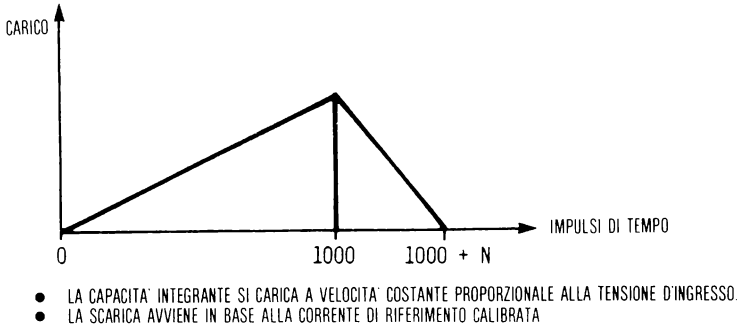


Fig. 5-14: Temporizzazione per integrazione

La fig. 5-15 illustra un dispositivo monolitico che usa la *tecnica di integrazione della doppia pendenza* conosciuta anche come *QUAD-SLOPE*. Oltre ad integrare le tensioni note e quelle sconosciute, tale dispositivo integra anche le inesattezze dovute a dispersioni ed errori dovuti alla presenza della massa. La tecnica della doppia pendenza dà risultati di alta precisione. Tale precisione è raggiunta a spese del tempo necessario per convertire il segnale analogico. Quindi tale conversione è *lenta* in rapporto alla precedente tecnica delle successive approssimazioni, ma risulta *più accurata.*

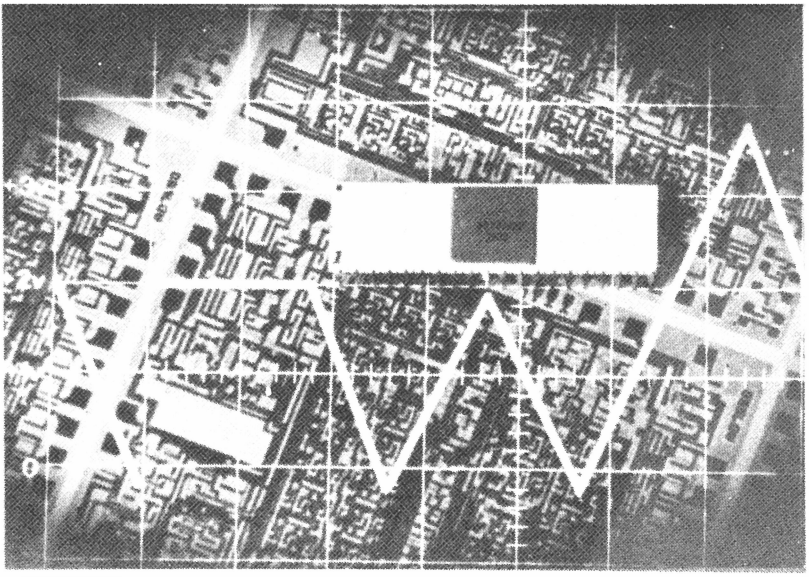


Fig. 5-15: Convertitore monolitico «quad slope»

CONFRONTO DIRETTO

Il metodo del confronto diretto è usato solo dove è richiesta una *estrema velocità*. Generalmente in tali applicazioni sono necessari meno di 5 bit di risoluzione. La circuiteria contiene 2^{n-1} comparatori (dove n è il numero dei bit della parola binaria in uscita). Vediamo come funziona tale metodo.

Supponiamo di avere un convertitore per confronto diretto a 3 bit. L'ingresso deve essere misurato in termini di 8 livelli. La fig. 5-16 illustra la struttura del nostro convertitore. I sette comparatori stabiliranno se la tensione di ingresso è maggiore o minore di ognuno degli otto possibili valori di riferimento. Per esempio, se tutti i comparatori al disotto del quinto sono in «on» e tutti quelli sopra sono in «off», allora il codificatore di priorità codificherà gli otto bit in un numero binario a 3-bit, 100_2. Altri ingressi verranno codificati in un'altra rappresentazione a 3-bit.

Tale sistema fornisce una risoluzione di cinque bit in meno di 100 ns per conversione.

A causa dei molti comparatori richiesti, delle tensioni di riferimento, il complesso schema per le priorità, questo metodo risulta essere il più costoso se si va oltre 3-bit di risoluzione.

Tuttavia l'AMD ha annunciato un dispositivo monolitico a 4-bit per meno di 50 $ il quale esegue questo confronto diretto in meno di 50 ns.

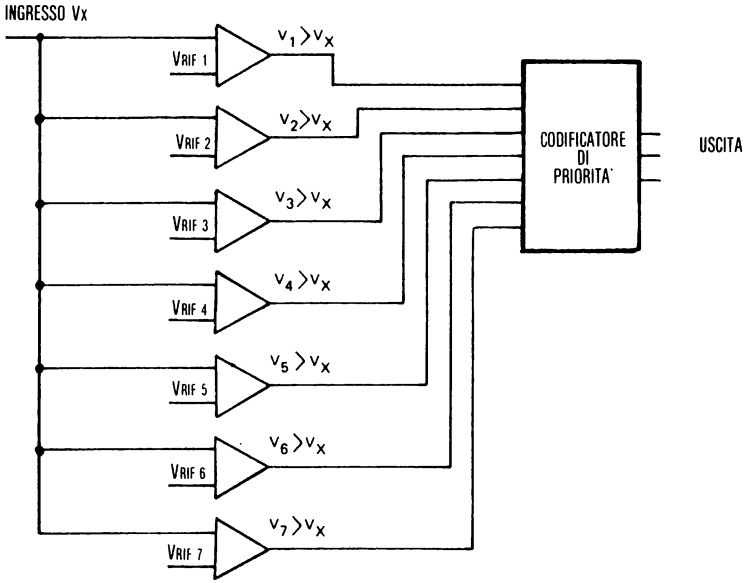


Fig. 5-16: Convertitore a confronto diretto

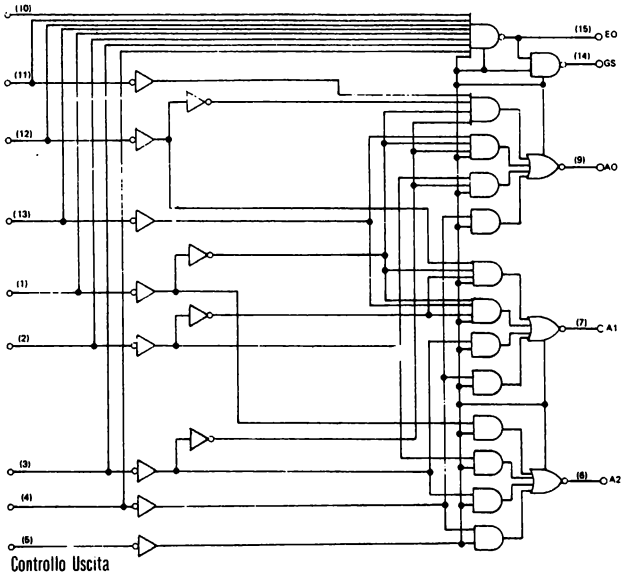


Fig. 5-17: Codificatore di priorità da 8 a tre

DISPOSITIVI TIPICI

La figura 5-18 mostra esempi di dispositivi di conversione e le tecnologie usate per implementarli. In generale più è veloce la conversione e più è costosa; più è accurata e più è costosa e più componenti esterni richiede e più è costosa.

COSTRUTTORE	TIPO #	RISOLUZIONE	VELOCITÀ	CONVERSIONE	COSTO
National	MM5357	8	40us	SA	$ 10
PMI	AD-02	8	8us	SA	
Analog Devices	AD7570	10	18us	SA	$ 70
Datel	ADC-EK 12B	12	24ms	Integrating	
Analog Devices	AD7550	13	40ms	Integrating	$ 25
National	ADC0816	8	114ms	SA	$ 20*

Fig. 5-18: Convertitori A/D

SOMMARIO A/D

Le tre tecniche viste di conversione A/D sono tutte disponibili su moduli monolitici LSI. I difetti delle tre tecniche sono facilmente individuabili. La conversione diretta è veloce ma poco precisa. La conversione per approssimazioni successive è mediocre sia in velocità che in risoluzione. La conversione a pendenza doppia è la più precisa ma è lenta. Infatti incide il tempo necessario per rendere stabile il segnale analogico, il tempo necessario per convertirlo, determinare la massima frequenza di campionamento e le richieste per il circuito di campionamento e stabilizzazione.

SVILUPPO DI UN SISTEMA ANALOGICO-DIGITALE DI RACCOLTA DATI

Se vogliamo interfacciare questi dispositivi di conversione al nostro sistema che raccoglie, analizza e controlla segnali analogici, dobbiamo sapere come usarli in modo ottimale. Normalmente occorre controllare più di un segnale. Ciò significa che occorrono più di un A/D e D/A. In alcuni sistemi occorre misurare molte centinaia di segnali analogici. Le tecniche usate nel progettare un sistema di costo interessante consistono nella multiplazione delle interfacce semplici.

Interfaccia per D/A

Il convertitore D/A richiede una parola digitale parallela che rimarrà tanto stabile quanto l'uscita analogica richiede. Questo è facilmente ottenibile per 8 o meno bit, poiché molti microcomputer hanno il controllo di trasferimento in uscita ad 8 bit.

---

* 16 canali completi Monolithic CMOS A/D Subsystem.

La fig. 5-19 mostra una interfaccia di questo tipo. Nel caso in cui il D/A ha più di 8 bit di risoluzione occorrono speciali tecniche.

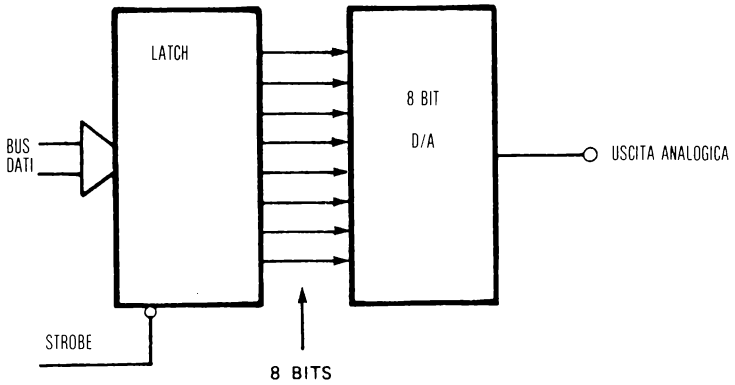


Fig. 5-19: Interfaccia del D/A all'uscita parallela

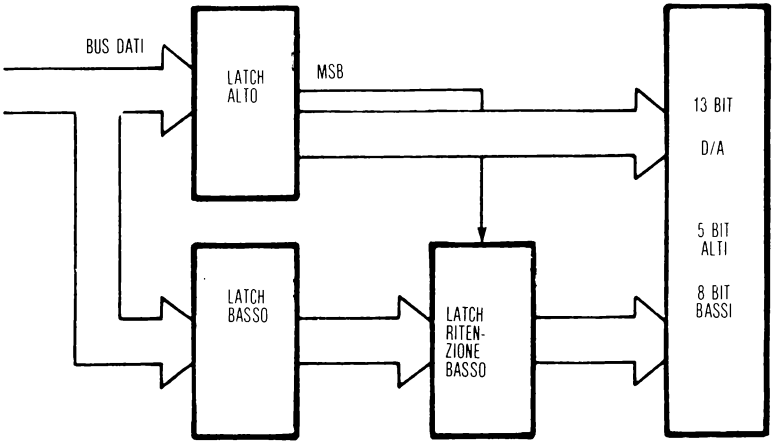


Fig. 5-20: Aggiunta di un controllore di trasferimento («latch»)

Per esempio consideriamo il caso di dover interfacciare un convertitore D/A a 12-bit. Se abbiamo due separati controllori di uscita ad 8 bit, usando 8-bit del primo e 4-bit del secondo vi sono problemi. Infatti quando il primo controllore di trasferimento è attivato, il D/A comincia immediatamente a convertire. Tuttavia alcuni microsecondi dopo si attiva il secondo controllore per completare il numero di bit richiesti dal D/A. L'effetto origina una *rapida variazione* del segnale di uscita del D/A a causa della variazione dell'ingresso. Tutti i bit di ingresso al D/A devono essere cambiati allo stesso istante per prevenire rapide variazioni dell'uscita. La fig. 5-20 mostra come un ulteriore controllore di uscita di più basso ordine, allo scopo

di impedire che i bit bassi cambino prima che siano cambiati i bit nel controllore di livello più alto.

Il byte-basso è inviato per primo ed il byte-alto per secondo con il bit più significativo uguale ad «1». Quando l'azione di bloccaggio temporaneo del controllore basso è effettuata dal bit ad «1» del controllore alto, i bit di basso ordine passano al convertitore ritardati dal ritardo del controllore stesso. Se questo ritardo è troppo lungo, un quarto controllore deve essere impiegato sul percorso dei bit alti per uguagliare il ritardo. I nuovi convertitori D/A offrono i *controllori di trasferimento su circuito integrato* per rendere più facili le interfacce.

Esempio di interfaccia di un D/A

La fig. 5-21 è lo schema per interfacciare un D/A al microprocessore SC/MP.

Il controllore di uscita ottale 74LS374 è usato per bloccare l'informazione mentre il D/A opera la conversione.

Anche se questo convertitore ha 12 bit di risoluzione in questo esempio ne sono stati usati solo 8. Gli ingressi non usati sono stati vincolati a +5V. Essi possono essere vincolati a +5V o a 0,0V in funzione dal tipo di codifica binaria usata.

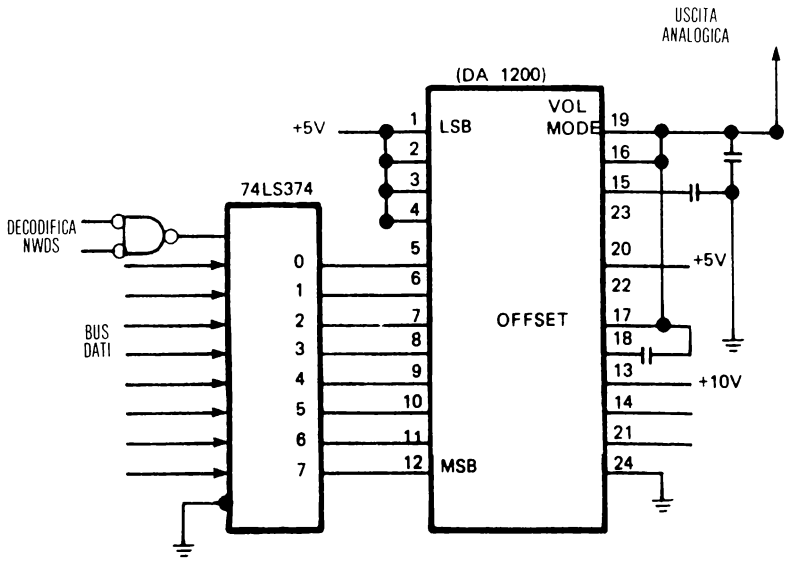


Fig. 5-21: Interfaccia del D/A SC/MP

Interfaccia per A/D

Come il convertitore D/A richiede una porta di uscita così un convertitore A/D richiede una porta di ingresso. In piú il convertitore A/D richiede un'uscita per iniziare una nuova conversione ed un ingresso per indicare quando è finita. Il tempo necessario per completare tale conversione può essere di 100 ms. Se si impedisse al processore di eseguire istruzioni mentre sta aspettando sarebbe uno spreco di tempo prezioso. Il convertitore A/D come dispositivo di ingresso dovrebbe operare in base a chiamate o ad interruzioni. Qui di seguito sono riportati cinque esempi di interfaccia per convertitori A/D.

Esempi di interfaccia per convertitori A/D

La fig. 5-22 illustra un convertitore A/D National MM 5357.
Questo dispositivo ha 8 bit per i dati in uscita, un ingresso per inizio conversione, un'uscita per fine conversione ed un ingresso per il segnale temporizzatore.

La linea di inizio conversione (SC) può essere attivata da un bit d'ingresso, o può essere terminata da un segnale di fine conversione (EOC). Se lo SC è vincolato all'EOC appena una conversione finisce comincia immediatamente un'altra. Il segnale di fine della conversione può essere connesso ad un bit della porta di ingresso, cosí esso può essere registrato. L'EOC può attivare un ingresso di una interruzione in funzione di considerazioni software.

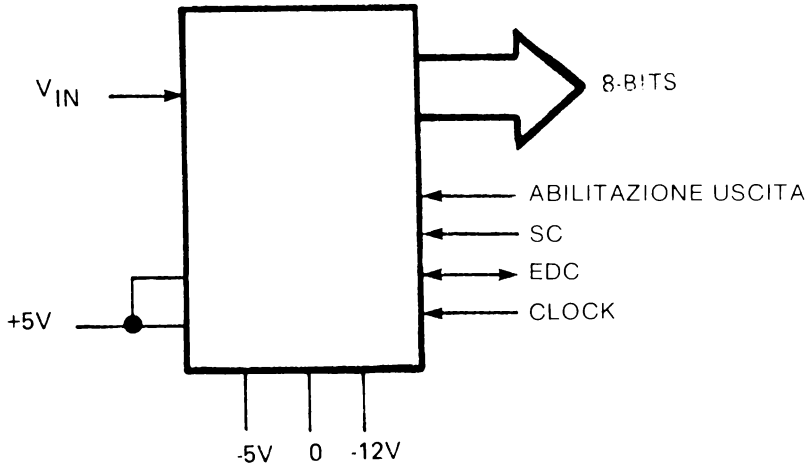


Fig. 5-22: Convertitore MM5357

La fig. 5-23 mostra l'uso di un convertitore A/D dove lo stato della conversione può essere letto sul bus dei dati, senza bisogno di avere pilotaggi di bus o una separata porta di ingresso. I dispositivi analogici AD 7550 sono in grado di fare questo usando dei *memorizzatori di uscita di tipo tri-state* che possono essere abilitati in maniera indipendente per il byte basso, per quello alto e per lo stato delle uscite.

Il solo segnale richiesto è il segnale per l'avvio della conversione che deve essere generato da un bit della porta di uscita.

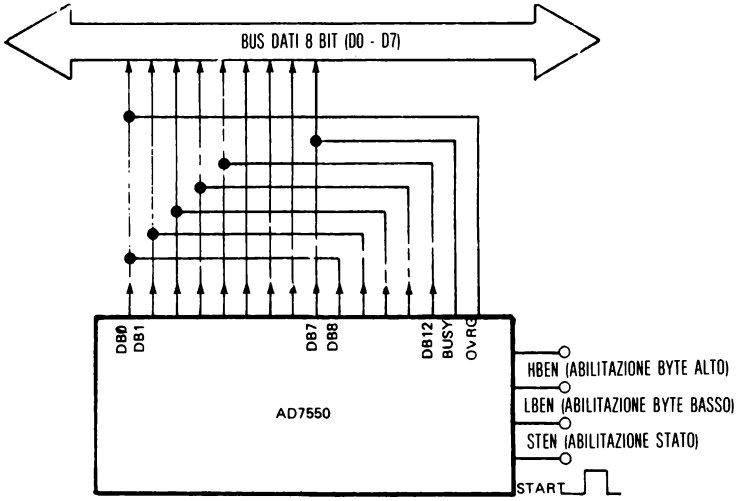


Fig. 5-23: Interfaccia dell'AD7550 della Analog Devices.

La maggior parte dei sistemi richiede piú di un ingresso analogico. Per fornire questi ingressi possiamo connettere un certo numero di convertitori A/D al bus e selezionarli con un *decodificatore.* Ogni ingresso dovrebbe avere il proprio convertitore A/D. La fig. 5-24 mostra lo schema per 4 convertitori in un sistema 8080. Il decodificatore 8205 seleziona i dati letti dalle porte (esadecimali) «F8», «F9», «FA» ed «FB». Le porte «FC», «FD», «FE» ed «FF» quando sono lette attivano le linee di inizio conversione nel corrispondente A/D.

La porta di ingresso «F0» è la parola di stato di fine conversione in cui i 4 bit piú bassi corrispondono alle uscite di fine conversione dei quattro convertitori A/D. Questa porta è interrogata dal programma per il controllo dei convertitori A/D.

La fig. 5-25 mostra una tecnica di approssimazioni successive effettuate mediante un comparatore D/A. L'adattatore di ingresso periferico (PIA) controlla il convertitore D/A ed ha come suo ingresso, l'uscita del comparatore.

Controllando il bit di uscita del comparatore si può dire se il byte di uscita sul PA0-PA7 è troppo grande o troppo piccolo. Il diagramma di flusso per tale metodo di successive approssimazioni può essere codificato in software ed usato per

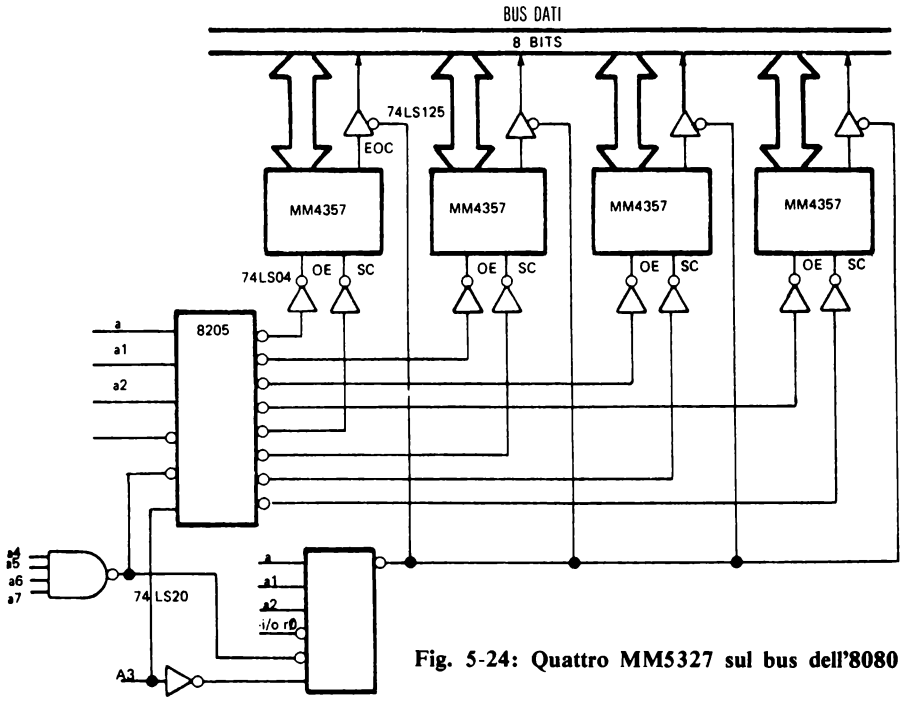


Fig. 5-24: Quattro MM5327 sul bus dell'8080

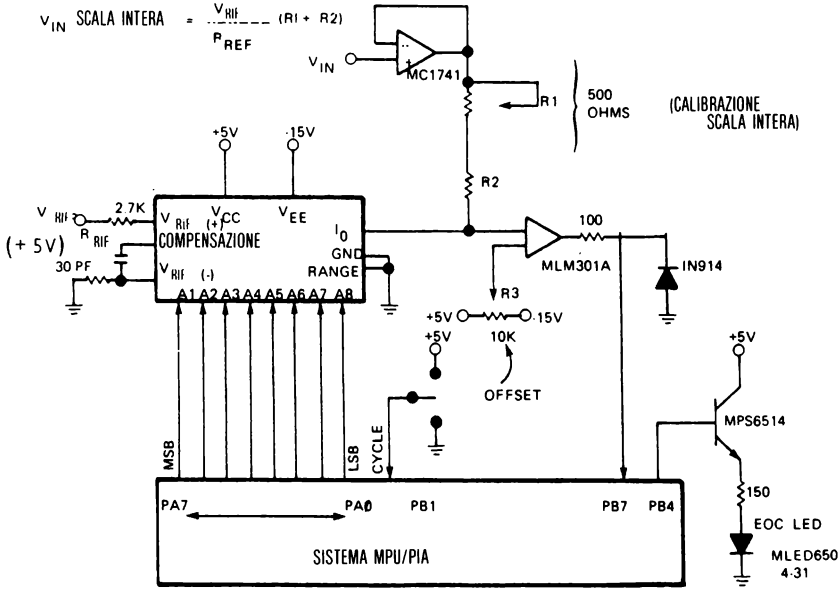


Fig. 5-25: Uso di un D/A per approssimazione successiva in conversione A/D

controllare lo svolgersi della conversione. Mediante questi cinque esempi abbiamo esaminato le tecniche per interfacciare l'A/D con un sistema a microprocessore.

Altri convertitori A/D hanno requisiti simili di interfaccia. Uno studio attento dei cataloghi e delle porte programmabili di ingresso-uscita risolve la maggior parte dei problemi di interfaccia verso l'A/D.

Esigenza di piú canali

Se sorge l'esigenza di piú canali di ingresso analogico oppure è troppo dispendioso avere un singolo A/D per canale vi è un'altra possibilità. Piú canali possono essere aggiunti mediante l'uso di una *multiplazione-analogica*. Tale sistema funziona come un commutatore in maniera tale che l'A/D possa essere selezionato da piú ingressi campione. Ciò è illustrato in fig. 5-26. L'intero sistema conterrà anche altri componenti A/D. Esaminiamo ora un dispositivo ibrido della Burr-Brown che interfaccia direttamente il bus del 6800.

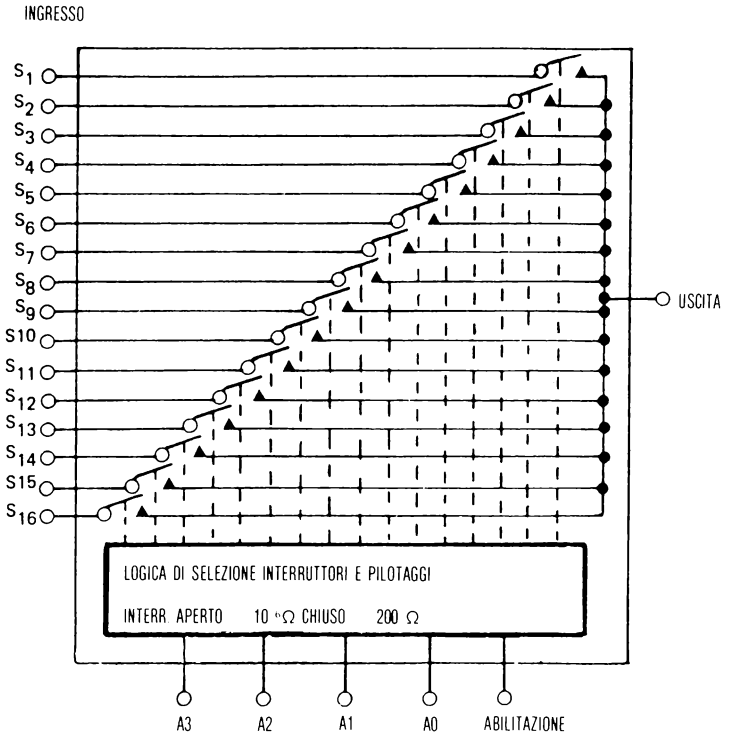


Fig. 5-26: Un multiplatore analogico

L'MP21

Il modulo MP21 contiene tutti i componenti necessari per fornire un sistema completo A/D per microprocessore 6800. Il diagramma a blocchi in fig. 5-27 illustra le funzioni interne del modulo MP21. Il modulo ha un multiplatore a 16 canali che può essere predisposto per fornire 8 ingressi differenziali oppure 16 singoli ingressi. Il numero di canale è selezionato dai quattro bit di indirizzo piú bassi e controllato con comando di lettura dalla logica di controllo.

Le apparecchiature di amplificazione provvedono alla conversione da differenziale a singolo (se richiesto) e può essere programmato da resistenze esterne per fornire guadagni e compensazioni diversi. Se richiesto un circuito di campionamento e mantenimento può essere inserito fra il multiplatore (multiplexer) e la strumentazione amplificatrice. Ulteriori multiplatori (multiplexer) possono essere aggiunti in questo punto per aumentare il numero dei canali di ingresso. Il cuore della unità è il convertitore A/D 8 bit che opera la conversione. La fine-conversione interrom-

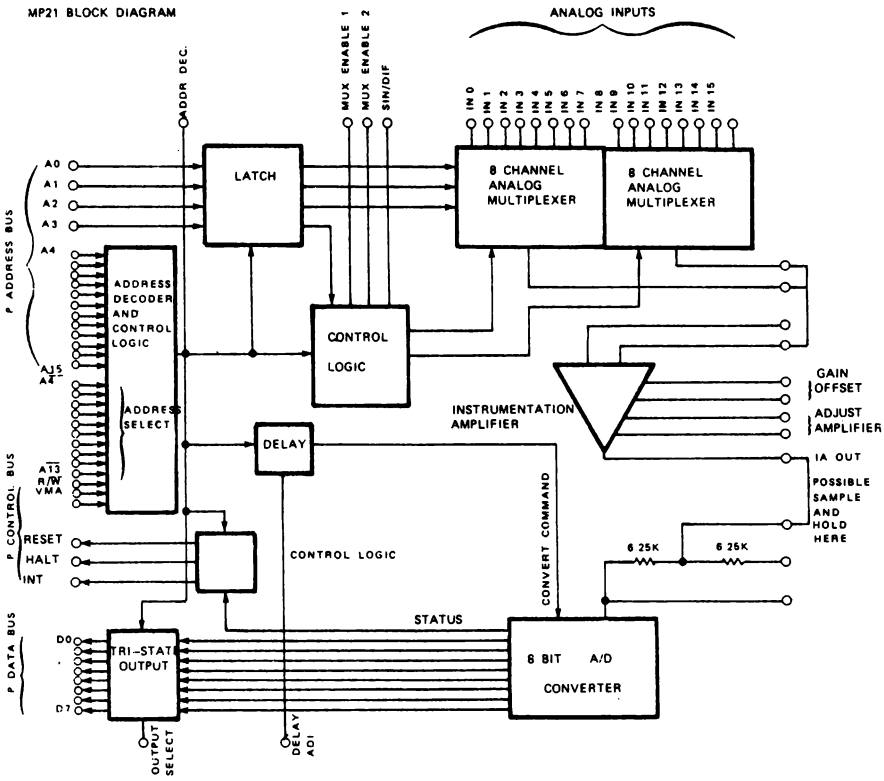


Fig. 5-27: Schema dell'MP21

pe il 6800 per mezzo della logica di controllo delle interruzioni interne nel modulo ibrido.

La figura 5-28 indica i segnali necessari per una tipica applicazione che utilizza un processore 650X o 6800.

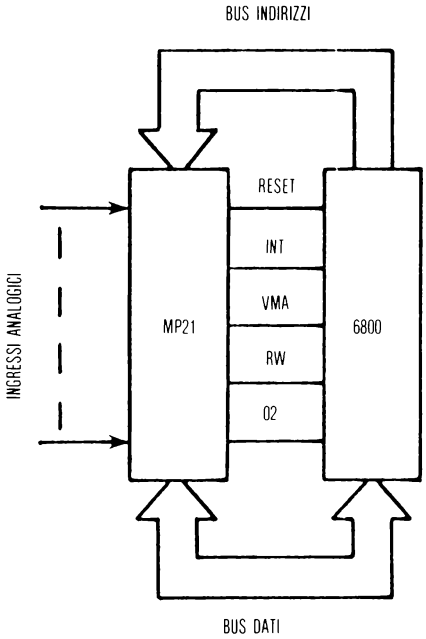


Fig. 5-28: Interfaccia tra il 6800 e il 650X

L'ADC0816

La National ha presentato un dispositivo monolitico CMOS con un multiplatore (multiplexer) a 16 canali, convertitore AD a successive approssimazioni e circuiti di pilotaggio del bus di tipo tri-state. Tutto questo è realizzato in un solo circuito integrato e costa 20 $ per grandi quantità. Con precisione fino ad 8 bit, l'ADC0816 costituisce un'unità completa di acquisizione dati in un unico integrato.

Tecniche per l'aumento della risoluzione

Vi sono due tecniche fondamentali per estendere la risoluzione di una conversione senza cambiare la precisione di base del convertitore A/D.

Tali tecniche sono il metodo *di variazione della scala* e quello di *compensazione.*

Variazione della scala

Se il segnale in ingresso è di 1,0 V e la massima scala della tensione in input può

arrivare a 10,0V, si deve incrementare in guadagno dell'amplificatore prima del convertitore A/D in modo che si possa sfruttare la risoluzione a fine scala dell'A/D stesso.

Aumentando il guadagno dell'amplificatore si possono misurare segnali piú piccoli piú accuratamente, se il segnale di input fosse stato di 20,0 V si sarebbe dovuto diminuire il guadagno dell'amplificatore d'ingresso in modo da attenuare il segnale d'ingresso stesso. Questo ci permette di misurare tensioni maggiori di quanto normalmente si potrebbe misurare. Da questi esempi risulta che la possibilità di variare la scala diventa una necessità evidente.

Si varia la scala per ottenere la massima informazione sul convertitore A/D.

Compensazione o fuori zero

Collegando una uscita di un convertitore D/A ad un ingresso compensatore prima dell'amplificatore si potrebbero correggere errori oppure si potrebbe compensare la tensione allo scopo di arrivare ad una precisione piú spinta. Se l'ingresso è 10,0 V e si è interessati a piccole variazioni intorno a questo valore, si può compensare l'ingresso con un valore di tensione uguale e contrario.

L'uscita dal convertitore D/A così compensato è −10,0V. Sommando insieme i due si ottiene un piccolo valore che dipende dalla differenza fra il D/A compensato e la tensione d'ingresso. Aumentiamo quindi il guadagno dell'amplificatore d'ingresso in modo che ogni differenza tra le tensioni di 10,0 V in ingresso e di 10,0V compensati possa essere così misurata con la massima precisione del convertitore A/D.

Riassunto delle tecniche di incremento della risoluzione

Con questi sistemi un convertitore A/D con risoluzione di 8 bit può essere migliorato per fornire molti piú bit di informazione sulle grandezze in gioco in forma codificata.

Nella figura 5-29 è riportato un esempio di forma codificata dell'informazione.

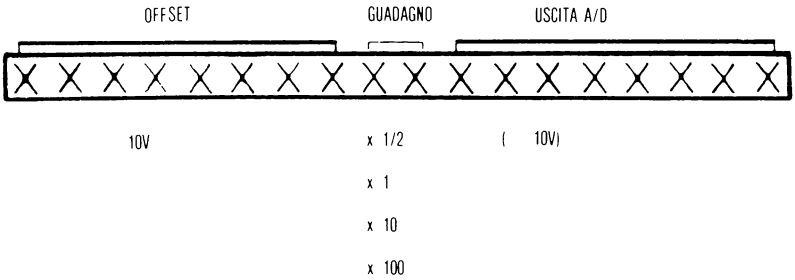


Fig. 5-29: Formati per segnali a scala, regolazione livelli per A/D

Naturalmente, la precisione dell'amplificatore e la compensazione di D/A devono essere tali da non introdurre errori propri nel processo di misura.

CONCLUSIONI

Il nostro microprocessore può ora essere usato per abilitare il trasferimento della informazione, la elaborazione e presentare in uscita l'informazione in una nuova forma nell'area analogica attraverso l'uso dei prodotti di conversione descritti.

Il convertitore D/A fornisce al microcalcolatore i mezzi per generare i segnali analogici. Il convertitore A/D fornisce invece i mezzi per misurare i segnali analogici, entrambi costituiscono la base di ogni sistema di conversione.

L'uso del campionamento e successiva raccolta dell'informazione, i multiplatori, le tecniche di scala e compensazione permettono di caratterizzare ogni segnale, processarlo e presentarlo nella forma richiesta.

Ultimo problema può essere fornire segnali di interfaccia in una forma standard. Ciò sarà esaminato nel capitolo 6.

CAPITOLO 6

TECNICHE E STANDARD DI BUS

INTRODUZIONE

Connettere più di un modulo richiede una via di comunicazione. Ciascun modulo deve essere capace di colloquiare con l'ambiente. I componenti entro un modulo devono essere inoltre capaci di colloquiare tra loro.

Il problema della interconnessione dei componenti é stato indicato nei capitoli 2 e 3. Le tecniche di comunicazione tra moduli e tra sistemi é trattato in questo capitolo, sono infatti di seguito descritte le *tecniche di bus.*

Esistono due distinti tipi di bus: bus paralleli e bus seriali.
Per ciascun gruppo sono di seguito trattati:

- paralleli
  - bus S100 di microprocessore
  - bus di microprocessore 6800
  - bus di interfaccia generale IEEE - 488
  - sistema di interfaccia CAMAC IEEE - 583
- seriali
  - comunicazione sincrona EIA RS232C
  - comunicazione sincrona ed asincrona EIA - RS422&423
  - standard di informazione ASCII
  - comunicazione sincrona

I bus *paralleli* sono utili per comunicazione ad alta velocità tra moduli nel caso di bus di microprocessore, e per comunicazione tra sistemi nel caso dell'IEEE-488. Unica eccezione é il CAMAC: lo standard CAMAC copre tutte le comunicazioni dal livello dei componenti in sú.

I bus seriali richiedono un numero minore di linee, e sono usati per connettere terminali di comunicazione al sistema elaboratore. Terminali come ad esempio i CRT, le telestampanti, le telescriventi, e i componenti di raccolta dati remoti, hanno necessità di una forma di comunicazione seriale per bit.

Gli standard seriali fissano le velocità, le caratteristiche elettriche, e il formato dei dati. Ci sono fondamentalmente due tipi di standard: sincrono e asincrono. Lo standard asincrono é usato per velocità minori di 20.000 bit al secondo, mentre lo standard sincrono é usato per velocità maggiori di 10.000 bit al secondo. Nel campo sovrapposto, possono essere usati entrambi i tipi.

Alla fine di questo capitolo sarà descritto un esempio di una interfaccia S100 applicata ad un convertitore analogico - digitale.

BUS PARALLELI

I bus paralleli trasferiscono tutti i bit di informazione attraverso vie separate, contemporaneamente e in parallelo. Devono essere previste vie per il bus dei dati, vie per il bus degli indirizzi, e vie per il bus di controllo. Ciascun gruppo di linee contiene informazioni relative al ciclo attualmente in corso.

Un sistema tipico a microprocessore richiede otto vie per i dati, 16 per gli indirizzi, e da 5 a 12 per il controllo.

- Le 8 vie dei dati servono per tutti i trasferimenti in ingresso e uscita dal microprocessore.
- Le 16 vie di indirizzo determinano su quale porta di I/O o su quale posizione di memoria deve essere fatto il trasferimento.
- Le 5 vie di controllo fondamentali sono per i cicli di scrittura e di lettura, per validità di indirizzo, per interruzione, per richiesta DMA e per richiesta attesa.

In un tale sistema base, il bus di controllo sarà temporizzato come in figura 6-1.

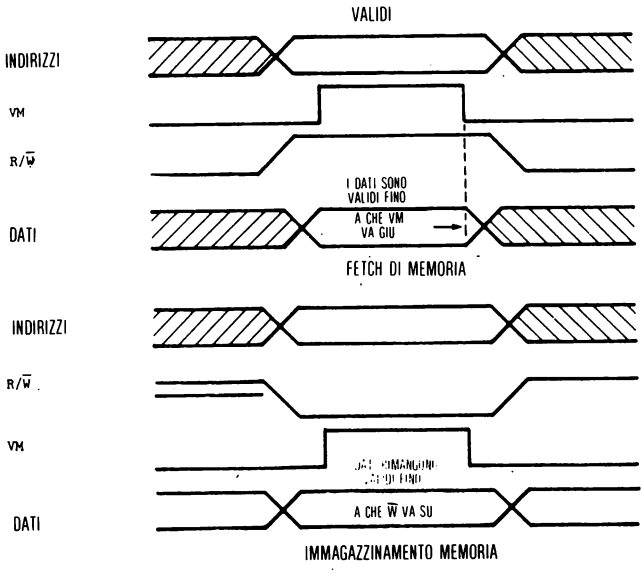


Fig. 6-1: Temporizzazione attraverso il bus di controllo

Questi 29 segnali sono tutti quelli necessari per gran parte dei bus paralleli semplici. La temporizzazione varierà, potranno essere usate vie separate di lettura e scrittura, ma tutte le funzioni operano in modo simile a quanto indicato.

Sistemi futuri richiederanno almeno 16 vie di dati e, probabilmente, 24 vie di indirizzo. Sono inoltre desiderabili linee di controllo addizionali per una gestione di ingresso - uscita flessibile.

IL BUS S100

Il mercato degli «hobby-computers» é esploso nell'Agosto 1976 alla conferenza di Atlantic-City. La presenza di una Società, tuttavia, ha assunto maggiore prominenza. La MITS, produttrice dei microcalcolatori Altair, ha usato 100 vie nel suo sistema basato sull'8080. Adesso ci sono più di 600 tipi differenti di piastre e sistemi che usano questo bus, prodotti da più di 100 costruttori. Nella conferenza citata, infatti, altri costruttori (in particolare la IMSAI) credettero che rendere i loro prodotti compatibili con tale bus avrebbe permesso loro di penetrare più facilmente in questo nuovo mercato.

I segnali del bus e le relative definizioni sono indicati nelle tabelle da 6-2 a 6-8.

Ci sono alcuni problemi in tale bus: vie di clock adiacenti a segnali di controllo, problemi di distribuzione delle terminazioni e distribuzione delle alimentazioni.

I segnali $\Phi 1$, $\Phi 2$, e il clock a 2 MHz sono vicini ad altri nove segnali di controllo. Tutti questi impulsi di clock hanno rapidi tempi di salita e di discesa e si verificano continuamente. Pertanto, tali segnali di clock sono facilmente accoppiabili con altre vie, a meno che non siano prese precauzioni mediante schermatura. Per la presenza del clock a 2 MHz il bus deve essere progettato con una immunità ad un rumore a 4 MHz, se non sono presenti altri segnali a quella frequenza.

Cosa succede se la piastra non é inserita correttamente? La possibilità che -18 volt si trovino dove sono richiesti 8 volt é una realtà. Se ciò capita, resta la speranza che non succeda nulla. Nel migliore dei casi potrà guastarsi il regolatore; nel peggiore, tutti i moduli connessi a +5 vanno fuori uso.

Idealmente la piastra potrebbe essere protetta contro inserzioni non allineate o inserimento rovesciato. Una idea potrebbe essere una organizzazione simmetrica delle terminazioni di potenza con interruzione dell'alimentazione in caso di inserzione non corretta; altra soluzione é una attenta distribuzione delle tensioni tra terminazioni poste a massa. Variazioni delle tensioni di alimentazione da modulo a modulo complicano il problema e riducono l'immunità al rumore. La soluzione é usare regolatori più costosi o eseguire delle regolazioni. Questa comunque non è la via migliore per risolvere il problema, mentre una distribuzione centrale della potenza comporta altri problemi.

Le linee di interruzione sono riservate per le richieste di interruzione ad una piastra controllo delle interruzioni connessa al bus. Non é definito un modo standard nell'uso di tali vie, e pertanto possono essere usati con il bus S100 anche lo Z-80, il 6502 (ed anche il 6800).

IL BUS S-100 (ALTAIR)

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
1	+8V	+8 Volts	Tensione non regolata sul bus, fornita alla piastra PC e regolata a 5 volt.
2	+18V	+18 Volts	Tensione positiva pre-regolata.
3	XRDY	Esterno pronto	Ingresso di esterno pronto verso la circuiteria pronta della piastra CPU.
4	VI0	Interruzione mediante vettore - via 0	
5	VI1	Interruzione mediante vettore - via 1	
6	VI2	Interruzione mediante vettore - via 2	
7	VI3	Interruzione mediante vettore - via 3	
8	VI4	Interruzione mediante vettore - via 4	
9	VI5	Interruzione mediante vettore - via 5	
10	VI6	Interruzione mediante vettore - via 6	
11	VI7	Interruzione mediante vettore - via 7	
12	*XRDY2	Esterno pronto 2	Una seconda via di esterno pronto simile a XRDY.
		* Nuovo segnale del bus per l'8080b	
13			
a	Da definire		
17			
18	STAT DSB	Status Disable	Permette che i tamponi per le 8 vie di stato siano tri-state.
19	C/C DSB	Command/control disable	Permette che i tamponi per le 6 vie di controllo/comando dello stato siano tri-state.
20	UNPROT	Unprotect	Ingresso al flip-flop di protezione memoria nella data piastra di memoria.

Fig. 6-2: Bus Altair

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
21	SS	Single step	Indica che la macchina è nel processo di realizzazione di un singolo passo (cioé che il flip-flop SS nel D/C è posizionato).
22	ADD DSB	Address disable	Permette che i tamponi per le 16 vie di indirizzo siano a tri-state.
23	DO DSB	Data out disable	Permette che i tamponi per le 8 uscite dati siano tri-state.
24	02	Phase 2 clock	
25	01	Phase 1 clock	
26	PHLDA	Conferma Hold	Segnale di uscita di controllo: comando del processore che appare in risposta al segnale Hold; indica che il bus dati e di indirizzo passerà allo stato HOLD dopo il completamento del ciclo di macchina corrente
27	PWAIT	Attesa	Segnale di controllo/comando del processore che appare in risposta al segnale READY che sta passando a livello basso; indica che il processore entrerà in una serie di stati di attesa di 0,5 μsec finché READY diventa nuovamente alto.
28	PINTE	Interrupt enable	Segnale di uscita di controllo/comando del processore; indica che le interruzioni sono abilitate, come determinato dai contenuti del flip-flop di interruzione interno alla CPU. Quando il flip-flop è posto a set (istruzione abilitazione interruzione) le interruzioni sono accettate dalla CPU; quando esso è resettato (istruzione disabilitazione interruzione) le interruzioni sono inibite.
29	A5	Via di indirizzo 5	
30	A4	Via di indirizzo 4	
31	A3	Via di indirizzo 3	
32	A15	Via di indirizzo 15	(MSB)
33	A12	Via di indirizzo 12	
34	A9	Via di indirizzo 9	
35	DO1	Via dati in uscita 1	

Fig. 6-2: Continua

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
36	DO0	Via dati in uscita 0	(LSB)
37	A10	Via dati in uscita 10	
38	DO4	Via dati in uscita 4	
39	DO5	Via dati in uscita 5	
40	DO6	Via dati in uscita 6	
41	DI2	Via dati in ingresso 2	
42	DI3	Via dati in ingresso 3	
43	DI7	Via dati in ingresso 7	(MSB)
44	SM1	Machine Cycle 1	Segnale di uscita di stato che indica che il processore è nella fase di prelievo del primo byte di una istruzione
45	SOUT	Output	Segnale di uscita di stato che indica che il bus di indirizzo contiene l'indirizzo di un componente in uscita e che il bus dati conterrà i dati in uscita quando PWR è attivo
46	SINP	Input	Segnale di uscita di stato che indica che il bus di indirizzo contiene l'indirizzo di un componente di ingresso e che dati in ingresso potranno essere posti nel bus dati quando PDBIN è attivo
47	SMEMR	Memory read	Segnale di uscita di stato che indica che il bus dati sarà usato per la lettura di dati da memoria
48	SHLTA	Halt	Segnale di uscita di stato che conferma una istruzione di HALT
49	CLOCK	Clock	Uscita invertita di 02 CLOCK
50	GND	Ground	
51	+8V	+8 Volts	Ingresso non regolato al regolatore a 5 volt
52	-18V	-18 Volts	Tensione negativa pre-regolata
53	SSWI	Sense switch input	Indica che sta per aver luogo un trasferimento dagli switch di acquisizione. Tale segnale è usato dalla logica di schermo. Controllo per: a) abilitare i pilotaggi degli switch acquis.; b) abilitare i pilotaggi di ingresso dati della piastra di controllo/visualizz. (FD10-FD17); c) disabilitare i pilotaggi di CPU di ingresso dati (D10-D17).

Fig. 6-2: Continua

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
54	EXT CLR	External clear	Segnale di azzeramento per i componenti di I/O (Switch del pannello frontale posto a massa)
55	*RTC	Real-time clock	Il segnale a 60 Herz è usato come riferimento di temporizzazione dalla piastra di clock in tempo reale/interruzione mediante vettore
56	*STSTB	Status strobe	Segnale di campionamento in uscita fornito dal generatore di clock 8224. Scopo principale è campionare il «latch» di stato dell'8212 in modo che lo stato sia posto ad «1» appena possibile nel ciclo di macchina. Tale segnale è anche usato dalla logica di schermo/controllo
57	*DIGI	Data input gate 1	Segnale di uscita dalla logica di schermo/controllo che determina quale gruppo di pilotaggio di uscita dati ha il controllo del bus dati bidirezionale della piastra CPU. Se DIGI è HIGH hanno il controllo i pilotaggi della CPU; se è LOV hanno il controllo i pilotaggi della logica di schermo/controllo
58	*FRDY	Front panel ready	Segnale di uscita dalla logica D/C che permette che il pannello frontale controlli le vie di READY verso la CPU.
59 a 67	DA DEFINIRE		
68	MWRITE	Memory write	Indica che i dati presenti nel bus dati di uscita siano scritti nelle posizioni di memoria attualmente nel bus degli indirizzi.
69	PS	Protect status	Indica lo stato del flip-flop di protezione memoria nella piastra di memoria attualmente indirizzata
70	PROT	Protect	Ingresso verso il flip-flop di protezione memoria della piastra attualmente indirizzata

*Nuovo segnale del bus per l'8800 b.

Fig. 6-2: Continua

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
71	RUN	Run	Indica che il flip-flop 64/RUN è in posizione reset; cioé la macchina è in modo RUN
72	PRDY	Processor ready	Ingresso di memoria e di I/O verso la circuiteria di attesa della piastra CPU
73	PINT	Interrupt request	Il processore riconosce una richiesta di interruzione su questa via alla fine dell'istruzione corrente o mentre in halt. Se il processore è nello stato HOLD o il flip-flop di abilitazione interruzione è in posizione reset, il processore non prende in considerazione la richiesta.
74	PHOLD	Hold	Segnale di ingresso di controllo/comando del processore che richiede al processore stesso di entrare nello stato HOLD; esso permette che un componente esterno acquisisca il controllo del bus dati e di indirizzi appena il processore ha completato l'uso di questi bus per il ciclo attuale di macchina
75	PRESET	Reset	Ingresso di controllo/comando del processore; mentre attivato, il contenuto del contatore di programma è azzerato e il registro di istruzioni è posto a zero
76	PSYNC	Sync	Uscita di comando/controllo del processore; fornisce un segnale per indicare l'inizio di ciascun ciclo macchina
77	PWR	Write	Uscita di comando/controllo del processore; usato per controllo di scrittura in memoria o uscita I/O. I dati sul bus relativo sono stabili mentre PWR è attivo
78	PDBIN	Data bus in	Uscita di comando/controllo del processore. Indica alla circuiteria esterna che il bus dati è nel modo ingresso
79	A0	Via indirizzi 0	(LSB)

Fig. 6-2: Continua

NUMERO DI PIN	SIMBOLO	NOME	FUNZIONE
80	A1	Via indirizzi 1	
81	A2	Via indirizzi 2	
82	A6	Via indirizzi 6	
83	A7	Via indirizzi 7	
84	A8	Via indirizzi 8	
85	A13	Via indirizzi 13	
86	A14	Via indirizzi 14	
87	A11	Via indirizzi 11	
88	DO2	Via dati in uscita 2	
89	DO3	Via dati in uscita 3	
90	DO7	Via dati in uscita 7	
91	DI4	Via dati in ingresso 4	
92	DI5	Via dati in ingresso 5	
93	DI6	Via dati in ingresso 6	
94	DI1	Via dati in ingresso 1	
95	DI0	Via dati in ingresso 0	(LSB)
96	SINTA	Interrupt acknowledge	Segnale di uscita stato; segnale di conferma alla richiesta di interruzione
97	SWO	Write out	Segnale di uscita di stato; indica che l'operazione nell'attuale ciclo di macchina sarà una funzione di WRITE in memoria o di uscita.
98	SSTACK	Stack	Segnale di uscita di stato; indica che il bus degli indirizzi memorizza l'indirizzo caricato sullo stack dal puntatore di stack
99	POC	Power-on clear	
100	GND	Ground	Massa

Fig. 6-2: Continua

Altri segnali sono quelli di controllo. Il bus S100 ne ha in numero maggiore di quanti ne usa qualsiasi sistema, e soffre del fatto di essere stato progettato prima che fosse disponibile per l'8080 il modulo controllore di sistema. Pertanto la presenza di molti segnali é dovuta al problema iniziale della Intel della limitazione del numero delle terminazioni, come discusso nel capitolo 2. Naturalmente, é necessario disporre di un nuovo bus S100, riducendo il numero di segnali a un valore più aderente alle attuali esigenze. Ciò probabilmente non avverrà mai. *Uno standard può essere migliorato: ma esso non lo é: questo é - il motivo per il quale esso é uno standard!*

Il bus S100 é di uso pratico, ed é facilmente utilizzabile in gran parte delle applicazioni. I problemi indicati possono essere evitati quando saranno considerati nuovi schemi di bus nei prossimi anni per sistemi futuri.

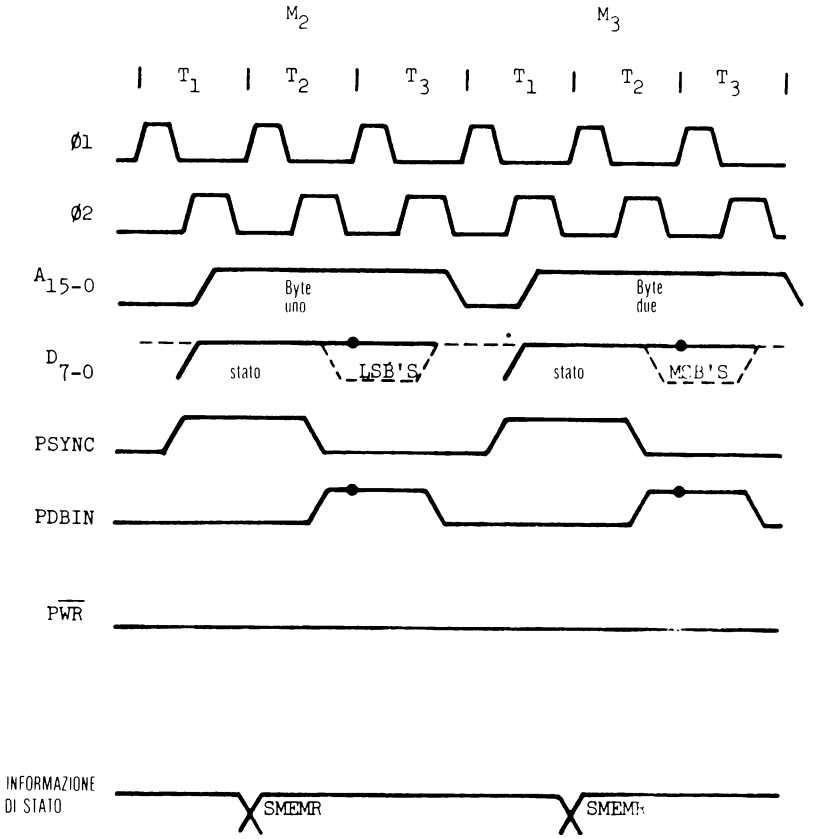


Fig. 6-3: Ciclo di lettura di memoria attraverso il bus S100

Il bus fornisce 8 vie per dati in ingresso, 8 vie per l'uscita, 16 di indirizzo, 3 di alimentazione, 8 per interruzione e 39 per controllo. Altre terminazioni sono lasciate per usi futuri.

Il *bus dei dati* é stato cambiato dal normale bus bidirezionale dell'8080 a due bus dati monodirezionali. Uno per dati in ingresso verso il microprocessore, ed uno per dati in uscita dal microprocessore. Spesso non é un vantaggio reale disporre di tale organizzazione perché molte periferiche connettono insieme i due bus. Unico inconveniente é però la necessità di disporre di altre otto terminazioni.

Il *bus degli indirizzi* é il tipico insieme delle 16 vie di indirizzo interfacciate da una memoria tampone.

Le *terminazioni di alimentazione* sono molto interessanti. Esistono due filosofie per la distribuzione della potenza: produrre tensioni regolate in una posizione di

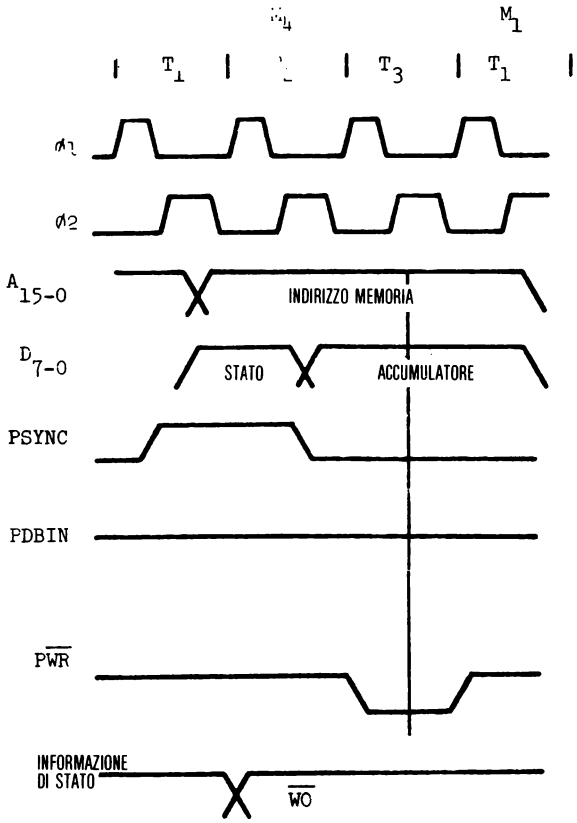


Fig. 6-4: Ciclo di scrittura in memoria attraverso il bus S100

controllo e distribuire la potenza, o regolare localmente in ciascun modulo di sistema. L'Altair ha scelto il secondo. Questa é una buona scelta perché la distribuzione della potenza ai moduli é semplificata, ed é ridotto il rumore per accoppiamento mutuo tra i moduli. È una scelta più costosa perché i regolatori costano molto di più di un unico buon regolatore e comporta la presenza di tensioni di alimentazione non perfettamente eguali tra i moduli.

Il progetto di una periferica compatibile con il bus S100 é discusso nell'esempio alla fine di questo capitolo. Nelle figure 6-3 e 6-4 sono indicate le temporizzazioni necessarie per i cicli di scrittura e lettura. Sono indicate le temporizzazioni basilari dell'8080, e i segnali fondamentali usati per il trasferimento. È da osservare l'importanza dei segnali $\overline{PWR}$ e PDBIN. Questi due segnali controllano la direzione dei dati nel bus: prelievo o memorizzazione. Oltre che dai segnali di stato, tutti i trasferimenti con la memoria possono essere identificati mediante queste poche vie.

IL BUS DEL SISTEMA 6800

È qui descritto il bus del sistema 6800 Altair-680B. Questo bus é stato impostato bene, in relazione ai problemi del bus S100.

Il sistema dispone di 8 vie di dati bidirezionali, 16 vie di indirizzo monodirezionali, e nove vie di controllo.

I bus per dati e indirizzi sono quasi identici a quelli presenti in altri sistemi. Le vie di controllo contengono il numero minimo di vie necessarie. Sono anche presenti: il clock $\Phi 2$, il reset, l'halt, R/$\overline{W}$, VMA, DBE, R/$\overline{W}$-P, BA e TSC. Essi sono descritti nella figura 6-5. Non sono indicati in tabella i segnali di richiesta interruzione IRQ e NMI. Essi sono presenti nel bus di controllo.

I segnali presenti in tale bus sono chiari, concisi per il prelievo e la memorizzazione dell'informazione. Questo é l'esempio di un progetto ben impostato. Sfortunatamente non sono presenti in questo bus i segnali $\Phi 1$ e , quello di pilotaggio primario e di $\Phi 2$, forse per ridurre l'immunità al rumore. Un buon clock isolato ad alta velocità é necessario in molte applicazioni senza necessità di costosi e non indispensabili pannelli schermanti.

IEEE - 488 - 1975

Questo bus é stato sviluppato per connettere sistemi, piuttosto che moduli. Componenti come calcolatori, voltmetri, alimentatori, generatori di frequenza, ed altri possono essere equipaggiati con il bus 488. Esso é il risultato di tre anni di discussione entro la IEC (International Electrotechnical Commission). Nel 1974 la IEEE ha approvato la bozza, ed é risultato lo standard IEEE - 488 - 1975. La Hewlett-Packard ha avuto una influenza fondamentale nello sviluppo di tale bus e la tecnica di «handshake» usata é brevettata dalla Hewlett-Packard. Tutti i produttori di una interfaccia compatibile con la 488 devono comprare la licenza per usare la circuite-

Il bus di controllo di sistema consiste dei seguenti segnali:

CLOCK: Il clock di sistema è a 500 kHz asimmetrico, a due fasi distinte, che oscilla ai livelli di tensione di alimentazione. La fase 1 (01) è usata per il funzionamento interno del modulo. Tutti i trasferimenti hanno luogo durante la e 2 (02). Pertanto, il segnale 02 è usato entro il sistema per abilitare la memoria e le interfacce, come l'adattatore d'interfaccia per comunicazione asincrona (ACIA).

RESET: Questo segnale è usato per inizializzare il sistema dopo una condizione di assenza di potenza dovuta sia ad un avviamento iniziale che ad una caduta di potenza. Esso è anche usato per riinizializzare la MPU in qualunque momento dopo una accensione. Quando è rivelato un fronte positivo nell'ingresso RESET, che è causato da un reset manuale sul pannello frontale, la MPU inizia la sequenza di restart. Entro la sequenza di restart, il Contatore di Programma è caricato con il contenuto della posizione del vettore di reset (FFFE, FFFF), che contiene l'indirizzo di partenza del «monitor» di sistema.

HALT: La via halt è usata per il controllo esterno dell'esecuzione del programma. Quando essa è nello stato alto (RUN), la MPU preleva l'istruzione indirizzata dal contatore di programma e inizia la esecuzione del programma. Quando la via di halt è bassa è bloccata tutta l'attività della MPU. In tal caso il segnale Bus Disponibile (BA) va al livello alto e gli indirizzi di lettura/scrittura (R/W) insieme alle vie dei dati sono nello stato di alta impedenza. Con BA alto, sono abilitati gli indirizzamenti da pannello frontale e le funzioni di deposito di dati.

R/W: Controlla la lettura/scrittura e indica la direzione del trasferimento dei dati. Quando è nello stato alto (READ), il dato è letto entro la MPU dalla memoria e dalle periferiche. Quando è nello stato basso (scrittura), il dato è scritto nella memoria o nelle periferiche. Quando il processore è bloccato, R/W commuta allo stato off (alta impedenza).

VMA: L'uscita VMA indica alla memoria o alle periferiche, come la ACIA, che è presente sul bus un indirizzo stabile e valido.

DBE: L'ingresso DBE è il segnale di controllo a tre stati per il bus dati della MPU e abilita i drivers del bus del 6800 quando è nello stato alto. La fase due è usata per pilotare direttamente questo ingresso. Durante un ciclo di lettura della MPU i circuiti di pilotaggio del bus dati sono internamente disabilitati, cioè all'interno della stessa MPU.

R/W-P: Il segnale Primitiva di R/W è ottenuto ponendo in NAND il segnale R/W con 02. Tale segnale assicura che i dati sono sempre letti e scritti mentre il bus dati è abilitato e non durante intervalli nei quali i dati non sono validi.

Fig. 6-5: Bus di controllo del sistema

ria di handshake del bus (il bus é chiamato talvolta HPIB o Hewlett-Packard Interface Bus).

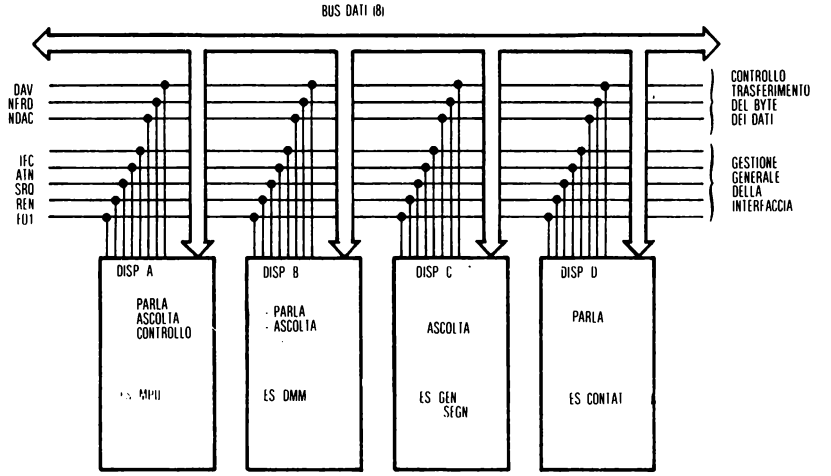


Fig. 6-6: Segnali del bus 488

Il bus di base realizza connessioni con componenti che svolgono una o più delle seguenti funzioni:

1. controllo di altre unità - *controllore*
2. prelievo di informazione dalla unità di controllo - *ascoltatore*
3. dare l'informazione alla unità di controllo - *produttore di parole*

Il bus consiste di otto vie di dati bidirezionali, tre vie di controllo del trasferimento di byte e cinque vie di controllo generale.

Le otto vie dei dati trasporteranno: comandi di componente (sono usati soltanto 7 bit), indirizzi e dati (8 bit).

Poiché il sistema *non ha indirizzi o bus di controllo completi,* il bus dei dati é utilizzato per eseguire tutte queste funzioni. Il resto delle vie controlla il modo di operare del bus dati e il modo nel quale esso é usato.

Le vie di controllo del trasferimento sono utilizzate per realizzare lo «handshaking» richiesto tra il componente che trasmette e quello che riceve.

Le ultime cinque vie controllano le condizioni generali del sistema. Esse sono: Attenzione, Cancella Segnali Interfaccia, Richiesta Servizio, Abilitazione Remota e Fine o Identificazione.

Attenzione, quando falso, indica che le vie dei dati contengono da uno ad otto bit di dati. Quando vero, il bus dati contiene un comando a sette bit o un indirizzo di sette bit.

Cancella segnali di interfaccia pone il sistema in uno stato noto. Esso é simile ad un reset di sistema.

Richiesta di servizio, quando é vero, indica all'unità di controllo che un componente richiede attenzione.

Abilitazione remota fissa il modo di operare di ciascun componente, insieme ad altri codici, in locale o in remoto.

Fine o identificazione é usato per indicare all'unità di controllo la fine del trasferimento dati.

La funzione di «handshaking» é usata quando componenti devono attendere che l'informazione sia disponibile. Una via dice, «come stai?». L'altra risponde: «bene, grazie. Io ho qualcosa per te». La risposta é: «dammela, io sono pronto». Il dialogo continua con «OK, eccola», e termina con, «grazie, piacere per l'incontro».

Nel nostro caso si dispone di tre vie: DAV (convalida dati sulla via dei dati), NFRD (non pronto per dati; se in stato vero indica che l'informazione é stata accettata dall'ascoltatore) e NDAC (dati non accettati; se in stato vero indica che il modulo di sistema è pronto ad accettare dati). In figura 6-7 è indicata la temporizzazione dello «handshake».

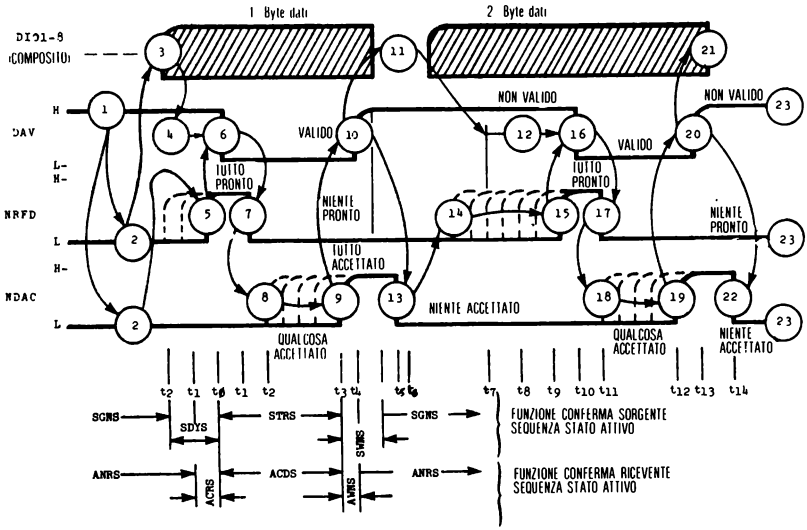


Fig. 6-7: Temporizzazione dello scambio controllato attraverso il bus 488

È da notare il modo nel quale l'ascoltatore accetta il trasferimento dei dati prima che sia iniziato il trasferimento successivo. Se esso appare complesso, esso lo é

realmente! L'uso di questo standard richiede una conoscenza completa di tutti gli stati permessi dal protocollo.

Nelle figure 6-8 e 6-9 sono indicati alcuni semplici esempi.

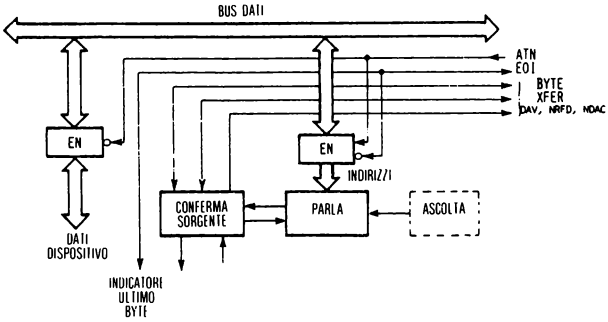


Fig. 6-8: Chi parla

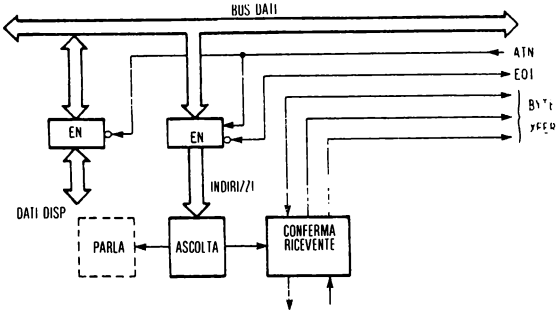


Fig. 6-9: Chi ascolta

Nell'esempio «parla», il controllore spedisce l'indirizzo e il comando di parlare a chi deve farlo, mediante l'uso di ATN e del bus dati. Una volta riconosciuto l'indirizzo ed il comando, chi deve parlare spedisce l'informazione all'ascoltatore, attraverso il bus dati, usando i segnali di «handshake». EOI indica infine che il trasferimento é stato completato.

L'esempio «ascolta» opera in modo simile. Il controllore spedisce l'indirizzo attraverso il bus dati, usando la via ATN come prima. In tal caso il comando successivo spedito é per il componente che deve ascoltare chi parla. Il trasferimento del dato, byte per byte, ha infine inizio usando il bus dei dati e i segnali di handshake. EOI infine indica che il trasferimento é completato.

In conclusione, il bus IEEE-488 rappresenta quasi un perfezionamento per i sistemi intelligenti di acquisizione dati. Man mano che i costruttori producono apparecchiature compatibili, lo standard diventerà ancora più diffuso di quanto lo é ora.

Infatti, il sistema microcalcolatore per applicazioni comuni della Commodore Business Machines é equipaggiato con una interfaccia bus IEEE-488. Ciò può indicare una nuova evoluzione nel calcolo comune così come nell'industria.

L'esempio di seguito sviluppato illustra come un sistema 6800 può essere interfacciato al bus 488. In figura 6-10 è indicato lo schema, che contiene la CPU 6800, il nuovo modulo di interfaccia della Motorola 68488, e il trasmettitore-ricevitore richiesto per il bus IEEE.

Il modulo permette che il 6800 sia facilmente interfacciato al bus IEEE. L'unità può essere sia un ascoltatore che un produttore di parole. In figura 6-11 è indicato un piccolo sistema 6800 con interfaccia GPIB. Il programma nella ROM preleva i dati dal canale seriale RS232C e li pone nel bus 488 quando richiesto. Quando i dati sono presenti sul bus 488, essi sono presentati in uscita attraverso l'ACIA sul canale seriale.

La figura 6-12 è una subroutine per la funzione di ascoltatore dell'uscita dell'ACIA. La routine assume che chi parla non lo faccia più velocemente di quanto l'ACIA non possa presentare in uscita. Ulteriori perfezionamenti possono essere aggiunti per tamponare i dati, convertirli in ASCII, se non lo sono già, aggiungere messaggi di EOI e così via.

Usando un po' di logica in più, può essere raggiunta dal sistema la funzione completa di controllore.

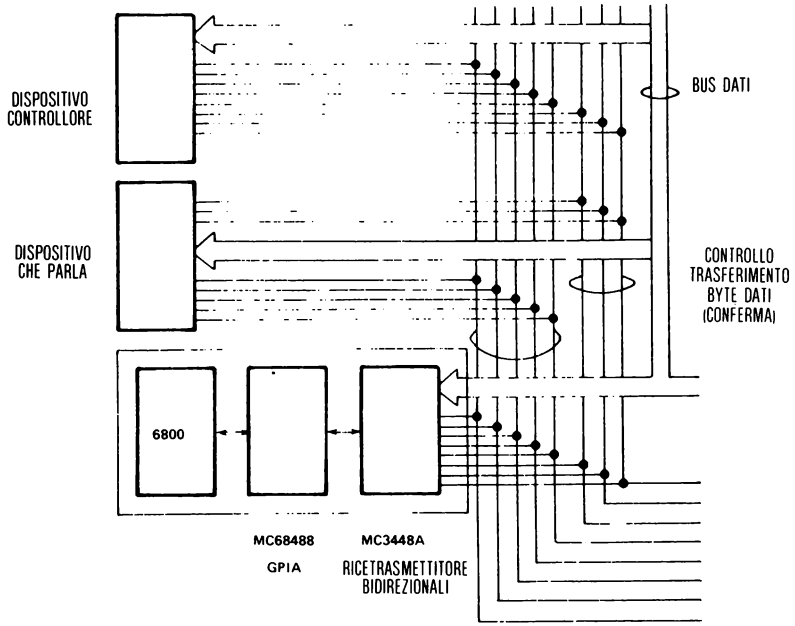


Fig. 6-10: Diagramma a blocchi del bus IEEE-488

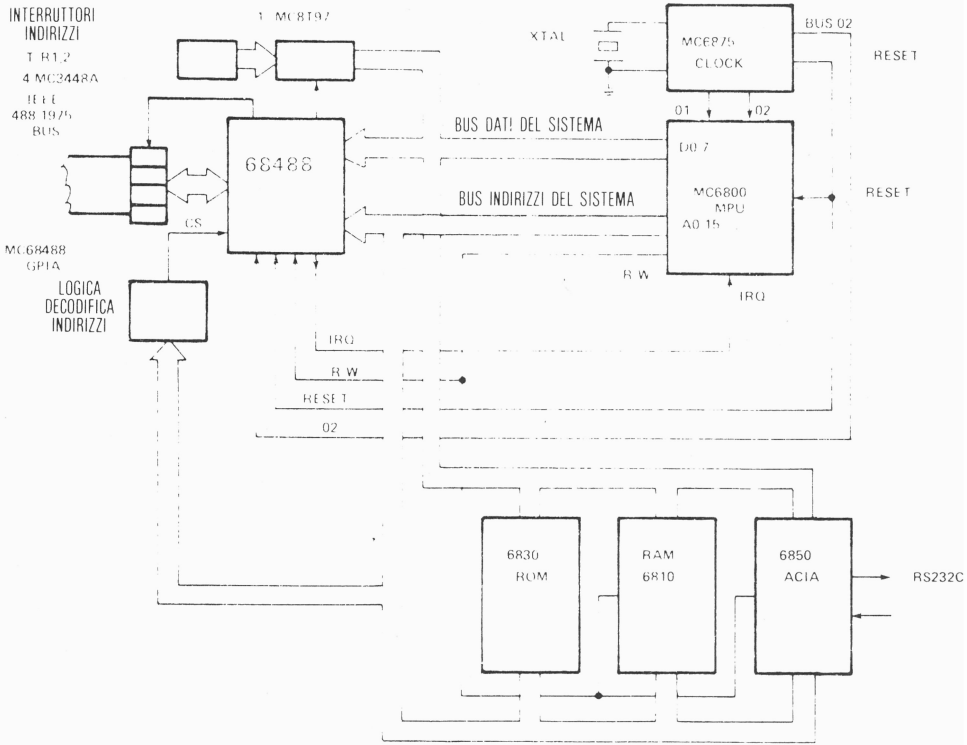


Fig. 6-11: Piccolo sistema 6800 della GPIB

Fig. 6-12: Programma software dell'ascoltatore

LDAA	#$XX	Carica quanto si trova nel controllo dell'ACIA nell'accumulatore A
STAA	$5008	Memorizza la velocità in baud, la parità e il numero di caratteri nel controllo dell'ACIA
LDAA	$5004	Leggi gli indirizzi dei componenti negli ADDRESS SWITCHES
STAA	$5004	Scrivi l'indirizzo nell'ADDRESS REGISTER
LDAA	#$00	Carica l'ACC con zeri
STAA	$5003	Ciò azzera il bit di reset
STAA	$5000	Maschera tutte le interruzioni (se desiderato) nell'INTERRUPT MASK REGISTER
STAA	$5002	Seleziona assenza di caratteristiche speciali nell'ADDRESS MODE REGISTER

NOTA: A tal punto il controllore indirizza il componente per «porsi in ascolto» nel seguente modo:
ENABLE ATN e spedisce mla (my listen address) sulle vie D101-8 che sarà X0100110 ($26). Poi DISABLE ATN. Una READ dell'ADDRESS STATUS REGISTER $5002 indicherà poi $86 ma (bit 7), LACS (bit 2) e infine LPAS (bit 1) sarà posto HIGH. Il componente è così pronto per LISTEN. B1 (bit 0) dello INTERRUPT STATUS REGISTER sarà LOW. B1 andrà HIGH per indicare che un byte di dati è disponibile nel DATA-IN REGISTER in $5007. La lettura della DATA-IN REGISTER resetterà B1 (bit 0).

LOOP1	LDAA	$5000	Carica l'ACC A con i contenuti dell'INTERRUPT STATUS REGISTER.
	TAP		Trasferisci i contenuti dell'ACC A al CONDITION CODE REGISTER.
	BCC	LOOP1	Esegui il loop finché il bit di riporto è settato. Ciò indica che B1 è settato in ROR
	BVS	LOOP2	Ramifica su LOOP1 se è settato fuori flusso, che indica END, bit 1 o che ROR è settato (cioé che il controllore ha spedito EOI).
	LDAA	$5007	Carica il DATA-IN REGISTER sull'ACC A. Ciò resetta B1
	STAA	$5009	Memorizza il byte dati nell'ACIA.
	INX		Incrementa il puntatore
	BRA	LOOP1	Ramifica nuovamente su LOOP1 e verifica se B1 è settato

```
LOOP2 INX          Incrementa il puntatore
      LDAA  $5007  Prende l'ultimo byte di dati dal REGISTRO
                   DATA-IN
      STAA  $5009  Pone l'ultimo byte nell'ACIA
      RTS          Fine della subroutine; il blocco è stato trasferito.
```

MAPPA DEGLI INDIRIZZI

INDIRIZZI ESADECIMALI	REGISTRI DELLO MC6848 (R/$\overline{W}$)
$5000	Stato di interruzione/maschera di interruzione
$5001	Stato di comando/—
$5002	Stato di indirizzo/modo di indirizzo
$5003	Comando ausiliario/Comando ausiliario
$5004	Commutazione indirizzo/indirizzo
$5005	Poll seriale/poll seriale
$5006	Comando passa oltre/Poll parallelo
$5007	Dati in ingresso/dati in uscita
$5008	Controllo dell'ACIA
$5009	Dati di ACIA

CAMAC

Lo standard IEEE-583 è quanto è conosciuto come il «Computer-Automated-Measurement-and-Control-Standard» o CAMAC. Esso copre anche standard relativi ad applicazioni del CAMAC.

Concettualmente il CAMAC si adatta a tutte le aree di interfacciamento degli strumenti. Esiste uno standard per le dimensioni del telaio e del connettore delle schede, esiste uno standard per l'alimentatore, ed esiste uno standard per il bus «dataway» (via dei dati). Esiste inoltre uno standard per il bus tra i telai: il «parallel highway» (via parallela ad alta velocità), uno standard per la comunicazione seriale tra i telai: la «serial highway» (via serie ad alta velocità).

È stato sviluppato per l'industria nucleare e tutto il dominio del CAMAC contiene specifiche rigorose. I sistemi CAMAC sono necessari se si vuol sviluppare secondo standard esattamente specificati.

Dimensioni fisiche

La figura 6-14 mostra una «cassa» CAMAC. La cassa è la sottounità di base del

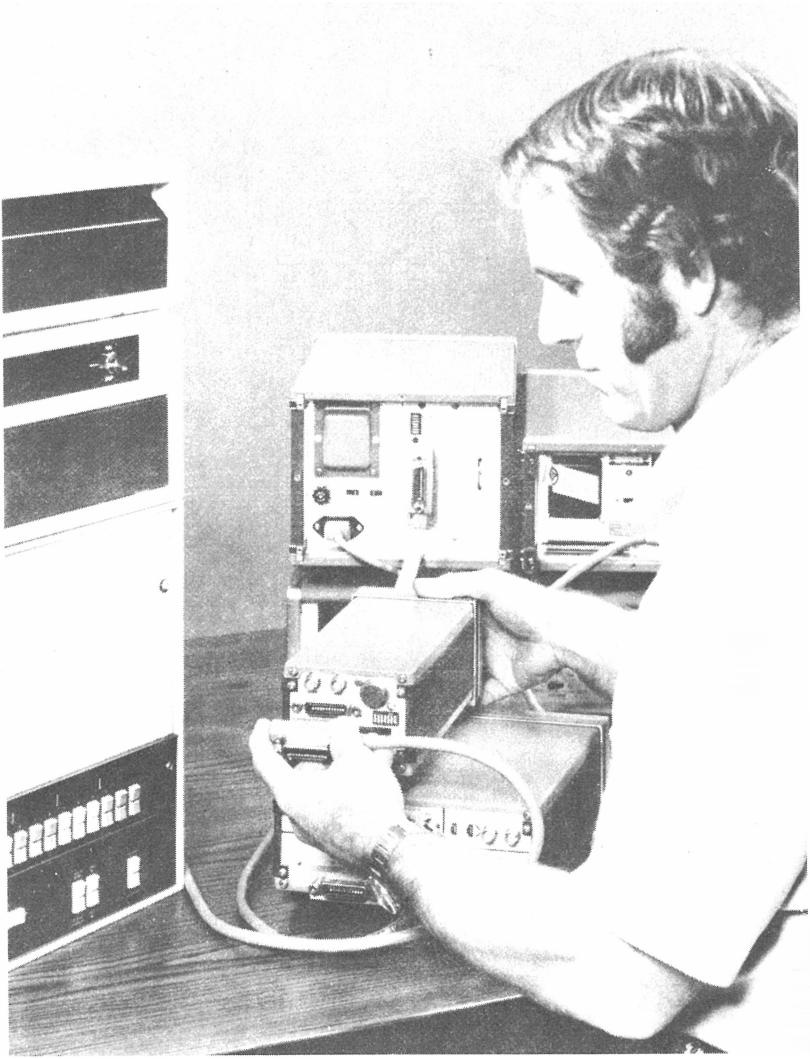


Fig. 6-13: Strumenti con IEEE-488 (Hewlett - Packard)

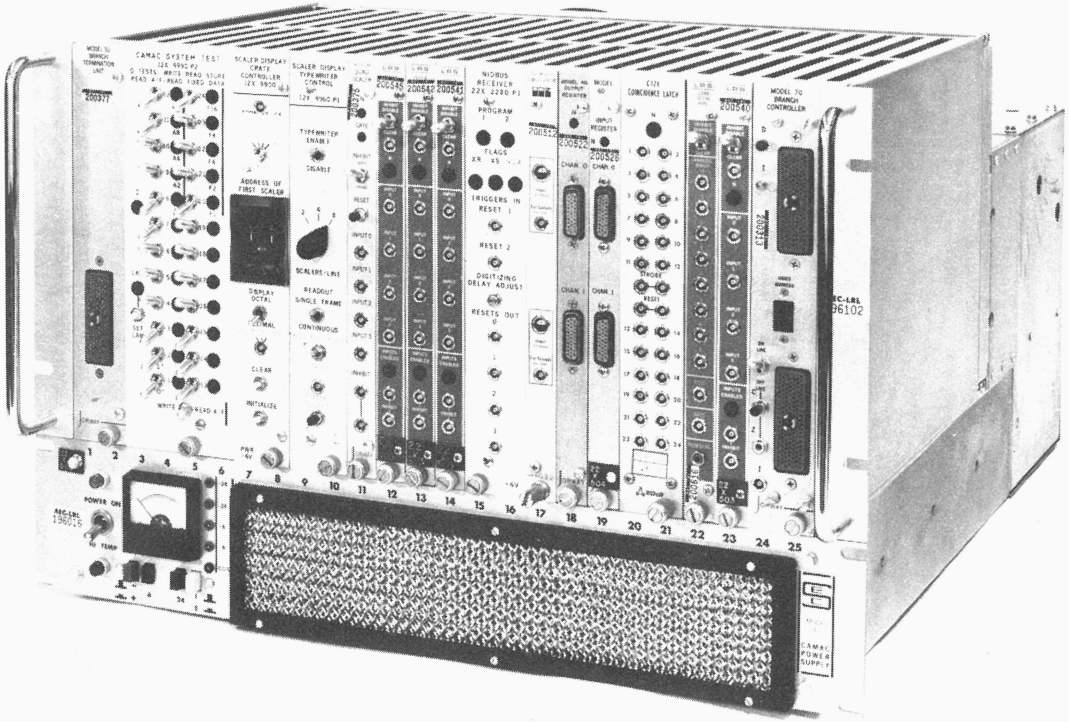


Fig. 6-14: Cassa CAMAC con i moduli

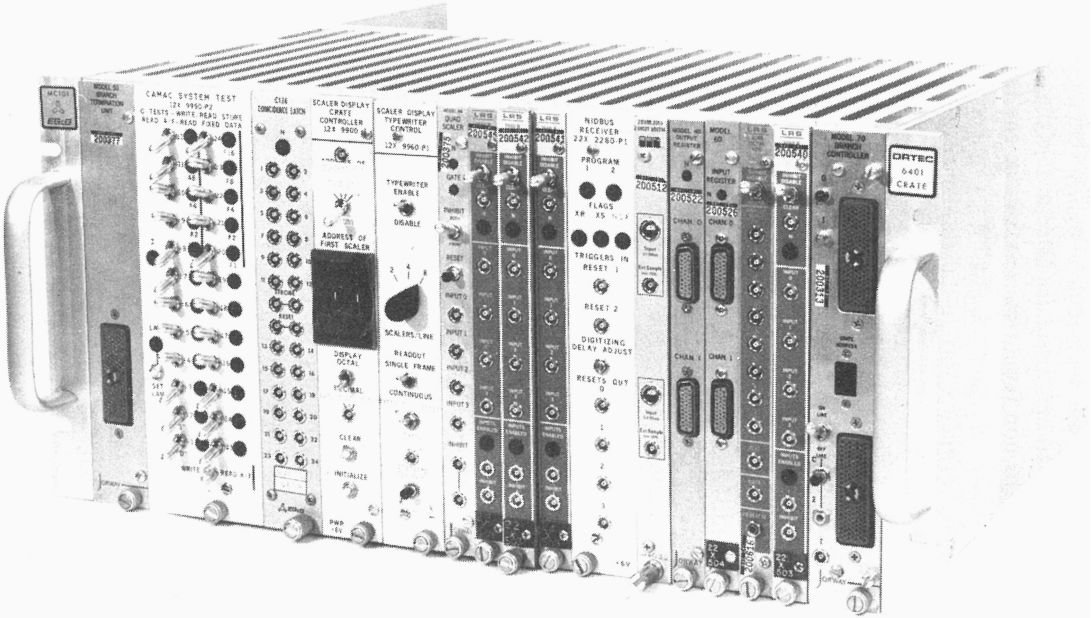


Fig. 6-15: Cassa e alimentatori

sistema. Essa contiene un controllore e fino a 24 interfacce di periferiche. La dimensione di ciascuna scheda e i tipi di connettore sono grandezze standard.

Alimentatore

L'alimentatore fornisce quattro tensioni, $\pm$ 6 volt e $\pm$ 24 volt. Lo standard specifica la stabilità, la regolazione e la soppressione dei transitori. È infatti da notare che l'alimentazione è la parte più importante di ogni sistema. Qualunque guasto alla alimentazione ha effetto su ogni parte del sistema. Pertanto, il CAMAC fa qualcosa che gli altri standard non fanno: esso garantisce che l'alimentazione sarà il problema meno importante nel sistema. La figura 6-15 mostra la cassa e l'alimentatore. (Le foto sono cortesemente fornite dal Lawrence Berkeley Laboratory).

Via dei dati

Il bus via dei dati della CAMAC consiste delle seguenti vie: tre controlli, cinque comandi, cinque indirizzi, ventiquattro letture, ventiquattro scritture, due temporizzazioni e quattro stati. Le vie sono descritte nella figura 6-16.

I tre controlli sono: inizializzazione, inibizione e cancellazione. Questi segnali sono usati per porre tutti i componenti connessi alla via dei dati in uno stato noto.

Le cinque vie dei comandi determinano la funzione da realizzare. Nello standard sono definite tutte le 32 possibili funzioni. Alcune funzioni sono per lettura, scrittura e trasferimento di stato. Altre sono lasciate per uso futuro o non definite.

Le 24 vie di lettura e scrittura costituiscono il bus dei dati. Se sono necessarie informazioni di indirizzo extra, i bus dei dati possono essere utilizzati per caricare ulteriori informazioni di indirizzo.

I 24 bit permettono un funzionamento efficiente mediante il trasferimento simultaneo di tre byte di 8 bit. Poichè alcuni sistemi contengono microprocessori, queste 24 vie possono trasferire i dati e gli indirizzi generati dal microprocessore. Poichè il trasferimento dei dati può avvenire a velocità tanto alta come 10^6 volte al secondo, questo bus richiede una banda maggiore rispetto ai bus fino ad ora descritti.

Il CAMAC può trasferire 24 bit x 10^6 volte al secondo, o 24 milioni di bit al secondo. Ciò è importante in applicazioni nucleari, dove un gran numero di dati devono essere trasferiti rapidamente durante ciascun esperimento.

I due segnali di temporizzazione forniscono l'informazione necessaria per indicare la validità dei dati.

Le vie di stato sono usate per indicare le richieste di servizio al controllore prodotte dalle periferiche sulle interfacce della via dei dati. Possono essere presenti 24 separate richieste in una singola cassa.

In conclusione, lo standard CAMAC realizza semplicemente un *concetto.* Esso copre tutti gli aspetti del problema della comunicazione. Esso include standard per il formato dei dati, per la comunicazione tra casse e convenzioni per il software.

A list of Dataway Signals Avaialble at Each of the Normal Stations 1 through 24 of a 25-Station CMAC Crate

Title	Designation	Use in Module
Common Control Signals		
Initialize	Z	Sets registers or control functions in a module to an initial state, particularly when power turned on.
Inhibit	I	Disables features for duration of signal.
Clear	C	Clears registers, or resets flip-flops.
Commands, addressed		
Function codes	F1,2,4,8,16	Carried on Dataway in binary code. Defines the function to be performed in a module during command operations.
Addressing		
Station number	N	Selects the module. There is an individual line from crate controller to each station.
Subaddress	A1,2,4,8	Also binary coded. Selects a location, within the module, to which the command is directed. There are 16 possible subaddresses.
Data		
Read bus	R1-R24	Transmits digital information from module to Crate Controller. Format is bit-parallel words, 24 bits maximum.
Write bus	W1-W24	Transmits digital information from Crate Controller to module. Format is same as for Read bus.
Timing		
Strobe 1 and Strobe 2	S1,S2	These strobes are generated by CC during every Dataway operation. Used by modules for timing acceptance of data or execution of features or an operation.
Status		
Look-at-Me	L	A signal from module to Crate Controller indicating request for service or attention. There is an individual line from each module to control station.
Q-Response	Q	A one-bit reply by module to certain commands issued by Crate Controller.
Command Accepted	X	Indicates the ability of a module to execute the current command operation.
Busy	B	Indicates a Dataway operation is in progress.

Fig. 6-16: Segnali di bus Dataway

STANDARD SERIALI

La trasmissione seriale richiede che soltanto uno o due fili trasportino tutti i segnali necessari tra i moduli o i sistemi. Essi sono spediti bit per bit.

Sono qui descritti gli standard di comunicazione sincrona e asincrona RS232C, RS422 e 423. Sono anche trattati standard per i dati, come l'ASCII e l'SDLC.

EIA – RS232C

La Electronics Industry Association (EIA) ha prodotto lo standard RS232C che copre le specifiche elettriche per la trasmissione seriale per bit, oltre alle specifiche fisiche. Esso definisce i segnali di «handshaking» usati per controllare le apparecchiature di connessione telefoniche standard e modulatori-demodulatori (modem) standard.

Elettronicamente, lo standard usa, per realizzare il trasferimento dell'informazione, impulsi di piú o meno 12 volt. Lo standard RS232C specifica una connessione a 25 terminazioni con i segnali indicati nella figura 6-17. Sono specificate tutte le 25 vie, ma nella tabella sono descritte soltanto le prime quindici:

– GROUND	
– XMIT DATA	(alla Unità di Comunicazione)
– REC DATA	(dalla U.C.)
– REQUEST TO SEND	(alla U.C.)
– CLEAR TO SEND	(dalla U.C.)
– DATA SET READY	(dalla U.C.)
– DATA TERMINAL READY	(alla U.C.)
– RING INDICATOR	(dalla U.C.)
– RECEIVED LINE SIGNAL DETECTOR	(dalla U.C.)
– SIGNAL QUALITY DETECTOR	(dalla U.C.)
– DATA RATE SELECTOR	(alla U.C.)
– DATA RATE SELECTOR	(dalla U.C.)
– TRANSMITTER TIMING	(alla U.C.)
– TRANSMITTER TIMING	(dalla U.C.)
– RECEIVER TIMING	(dalla U.C.)
+ DATI E SEGNALI SECONDARI	

Fig. 6-17: Segnali dell'EIA RS232C

Le vie secondarie forniscono i supporti per dati e controlli di un secondo canale seriale operante ad una velocità molto minore del canale primario. Il secondo canale è pertanto identico al primo, eccetto che per la velocità. Esso è difficilmente usato, e quando lo è, contiene informazioni di controllo per i modem connessi a ciascun estremo della linea di comunicazione.

Le vie dei segnali principali sono TRANSMIT DATA e RECEIVE DATA. Tali vie sono usate per spedire informazioni seriali tra due sistemi. I bit al secondo tra-

smessi hanno uno dei valori tra quelli standard indicati:

19.200	1.200	110
9.600	600	75
4.800	300	50
2.400	150	

Altre velocità sono occasionalmente usate. I terminali telescriventi operano a 110,150, o 300 bit/sec. I terminali CRT operano a velocità tipicamente superiori a 1200 bit/sec.

Molto spesso, i dati seriali sono trasmessi su linee telefoniche a condizionamento vocale. I dati devono essere prima modulati, in modo da poter essere trasmessi. Per velocità minore di 300 bit/sec la tecnica di modulazione usata è la FSK: «frequency-shift-keying» (commutazione di frequenza su chiave). La condizione «mark» o «1» logico è rappresentata da un tono di determinata frequenza, lo «spazio» o »0» logico è rappresentato da una frequenza diversa. Velocità di trasmissione maggiori di 300 bit al secondo usano la tecnica di modulazione di fase, per la mancanza di banda disponibile. Abbastanza spesso, le linee a condizione vocale sono troppo rumorose per comunicazione ad alta velocità.

Devono allora essere usate linee condizionate per dati.

Gli altri segnali sono usati per indicare lo stato della giunzione modem.

Segnali come: «request-to-send», «clear-to-send», «data-set-ready», «data-terminal -ready», sono usati a tale scopo.

La temporizzazione in figura 6-18 è rappresentata per indicare una tipica operazione di comunicazione. È da osservare che i segnali tra il modem e il terminale (o calcolatore) realizzano una forma di «hand-shake» simile a quella utilizzata in molti

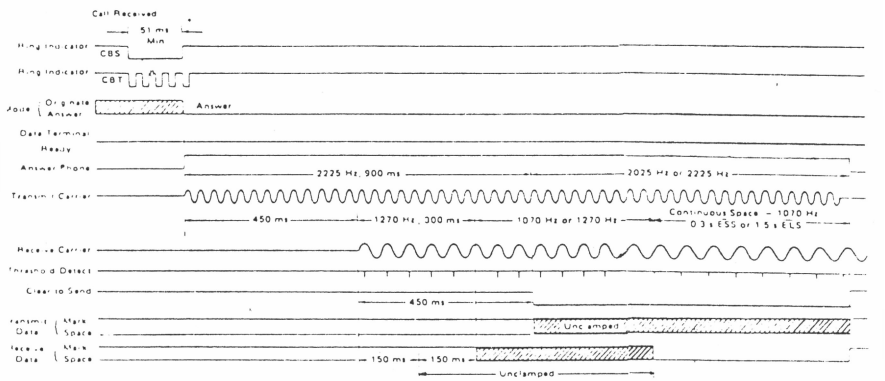


Fig. 6-18: Scambio di segnali di un modem attraverso l'EIA RS232C

bus-specialmente nello standard IEEE-488. La differenza è che in tal caso lo handshake è realizzato soltanto all'inizio e alla fine di un blocco di dati seriali.

Lo RS232C è popolare, infatti gran parte dei sistemi «time-sharing» su supporto telefonico usano questo standard nei loro sottoinsiemi di comunicazione. Uno standard simile è quello «*current loop*» (a ritorno di corrente). Esso è usato nelle telescriventi meccaniche. Una buona procedura da utilizzare è quella di convertire tutti i componenti a ritorno di corrente nello EIA-RS232C attraverso un convertitore da loop ad EIA. In tal modo tutte le comunicazioni sarebbero standardizzate. In figura 6-19 è rappresentato un tale convertitore per una teletype.

Questa è un'area nella quale i calcolatori, i modem, i terminali non rispettano completamente lo standard. Ponticelli presenti spesso permettono al terminale di operare come se tutte le condizioni per il passaggio dei dati fossero rispettate.

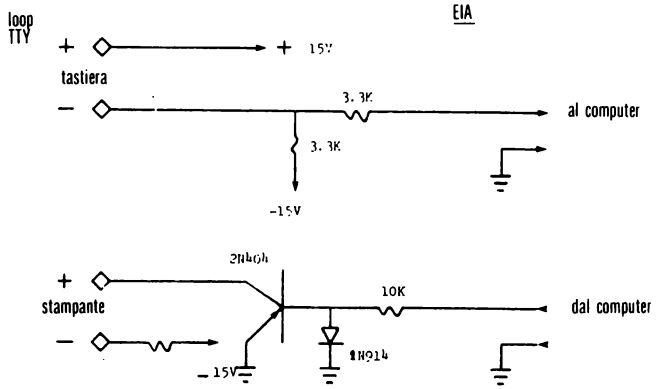


Fig. 6-19: Convertitore da segnali tipo ritorno corrente a EIA

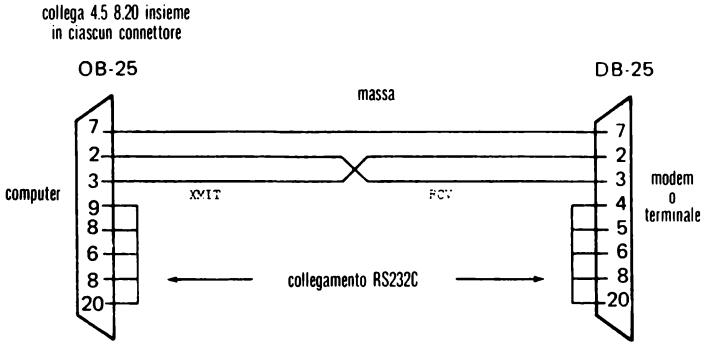


Fig. 6-20: Conversione ad anello di ritorno automatico

RS422 e 423

Lo RS232C trasmette i segnali mediante segnali in una unica direzione. Il «mark» o lo «space» sono rappresentati mediante tensioni tra due fili. Pertanto, la giunzione ha quattro fili, due di trasmissione e due di ricezione. Operare a quattro fili permette che la giunzione tra i componenti può essere piú lunga, per la maggiore immunità del canale differenziale. Nello stesso tempo la velocità di trasmissione può essere piú elevata per la maggiore immunità al rumore.

CHARACTERISTIC	RS232	RS422	RS423
MAXIMUM LINE LENGTH	100 ft.	5000 ft.	5000 ft.
MAXIMUM BITS/SEC.	2×10^4	10^6	10^5
DATA "1" = MARKING DATA "0" = SPACING	-1.5V → -36V +1.5V → +36V	VA > VB VA < VB	VA = - VB = +
SHORT CIRCUIT	100 mA	100 mA	100 mA
POWER-OFF LEAKAGE, MAXIMUM VOLT APPLIED TO UNPOWERED	300 μA	100 μA	100 μA
RECEIVER INPUT, MINIMUM	1.5V (single-ended)	100 mV (differential)	100 mV (differential)

Fig. 6-21: Confronto tra RS232C, RS422 e RS423

La figura 6-21 illustra differenze tra i tre standard. La figura 6-22 mostra i tipi di circuiti di pilotaggio e di ricezione usati per le linee. Gli standard RS422 e 423 non sono usati spesso per il già abbastanza diffuso uso della RS232C e la necessità poco frequente di usare velocità cosí alte e su distanze cosí lunghe.

Naturalmente i dati spediti possono essere codificati in diversi modi. Nei paragrafi successivi sono trattati i criteri di trasmissione sincrona, asincrona e gli standard dello scambio di informazioni.

COMUNICAZIONE ASINCRONA

I dati sono spediti *in modo asincrono* quando sono trasferiti treni di impulsi di eguale durata, senza l'informazione di clock. Quando i dati sono invece spediti con carattere di sincronizzazione presenti entro i blocchi di dati, essi sono spediti *in modo sincrono*, con un clock.

In figura 6-23 è indicata la struttura asincrona più comune, usata per gran parte dei CRT e delle telescriventi. Essa consiste di 10 (o 11) bit, come già indicato nel capitolo 4. Compongono un *carattere* un bit di start, otto bit dati e uno o due bit di stop. Gli standard piú popolari per i codici dei caratteri sono l'ASCII e lo EBCDIC.

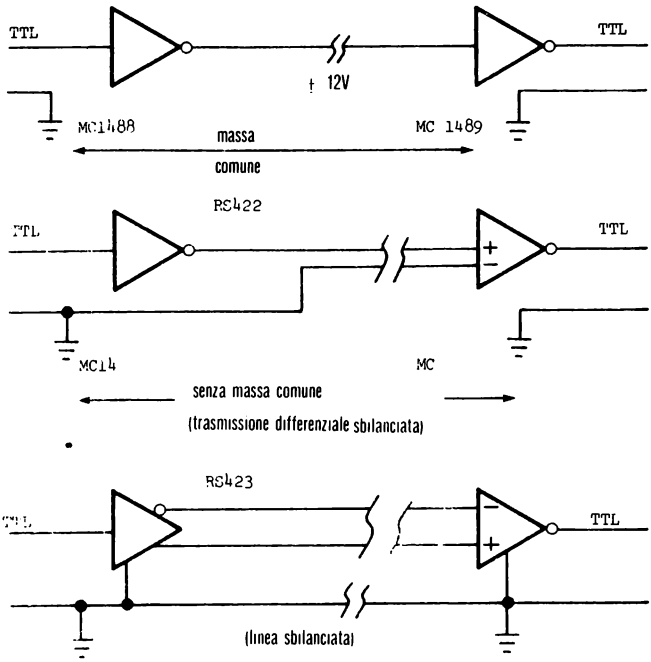


Fig. 6-22: Pilotaggi per la RS422 e RS423

ASCII significa «American Standard Code for Information Inter change».

Esso usa sette bit per codificare 128 caratteri possibili. Un ottavo bit può essere usato per la parità. Molti dei codici presenti sono usati per controllare le funzioni della giunzione. Codici come: «Begin text», «end of text», etc. sono usati per creare il formato e trasferire blocchi di caratteri.

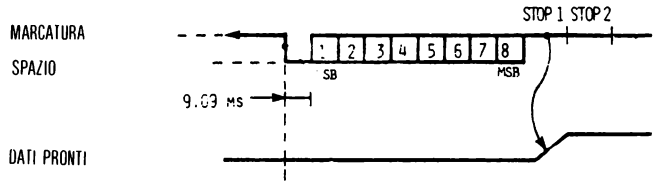


Fig. 6-23: Formato di dati seriali asincroni

Il codice EBCDIC è simile eccetto che i 128 codici sono codificati diversamente. Semplici ROM per conversione di codice convertono l'ASCII nell'EBCDIC e inversamente. Tali ROM hanno 8 ingressi: sette vie di indirizzo per i dati di ingresso, e una via di indirizzo per specificare la conversione realizzata. Esse hanno sette vie di uscita per il carattere convertito. La dimensione di tale ROM è di 256 byte di 7 bit per byte.

Questa è una ROM semplice rispetto agli attuali standard di sviluppo, ed economica nel prezzo e nella programmazione.

Chi usa l'EBCDIC? La IBM. Chi usa l'ASCII? Praticamente tutti gli altri. Esistono altri codici, come quello Baudot a cinque bit (ormai fuori uso), che può essere anche convertito mediante ROM di controllo.

Naturalmente, potrebbe essere usato un programma per operare la conversione da un codice ad un altro.

NUMERI DI BITS								0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
b_7	b_6	b_5	b_4	b_3	b_2	b_1	HEX 1 / HEX Ø	0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
			0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
			1	0	0	0	8	BS	CAN	(	8	H	X	h	x
			1	0	0	1	9	HT	EM	)	9	I	Y	i	y
			1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	11	VT	ESC	+	;	K	[	k	{
			1	1	0	0	12	FF	FS	,	<	L	\	l	¦
			1	1	0	1	13	CR	GS	-	=	M	]	m	}
			1	1	1	0	14	SO	RS	.	>	N	^	n	~
			1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Fig. 6-24: Tabella del codice ASCII

ACK	Acknowledge (conferma)
BEL	Bell (campanello)
BS	Backspace (passo in dietro)
CAN	Cancel (Cancella)
CR	Carriage return (ritorno carrello)
DC1	Direct control 1 (controllo diretto 1)
DC2	Direct control 2 (controllo diretto 2)
DC3	Direct control 3 (controllo diretto 3)
DC4	Direct control 4 (controllo diretto 4)
DEL	Delete (cancella)
DLE	Data link escape (dati di giunzione fuori codice)
EM	End of medium (fine del mezzo)
ENQ	Enquiry (interrogazione)
EOT	End of transmission (fine della trasmissione)
ESC	Escape (fuori codice)
ETB	End transmission block (fine blocco di trasmissione)
ETX	End text (fine testo)
FF	Form feed (Attivazione formato)
FS	Form separator (separatore di formato)
GS	Group separator (separatore di gruppo)
HT	Horizontal tab (tabulazione orizzontale)
LF	Line feed (avanzamento di linea)
NAK	Negative acknowledgement (conferma negativa)
NUL	Null (nullo)
RS	Record separator (separatore di record)
SI	Shift in (scorrimento interno)
SO	Shift out (scorrimento esterno)
SOH	Start of Header (inizio testata)
SP	Space (spazio)
STX	Start text (inizio testo)
SUB	Substitute (sostituisci)
SYN	Synchronous idle (ozio sincrono)
US	Unit separator (separatore unitario)
VT	Vertical tab (tabulazione verticale)

Fig. 6-25: Abbreviazioni della tabella del codice ASCII

Esadecimale	EBCDIC 8 bit	Esadecimale	EBCDIC 3 bit	Esadecimale	EBCDIC 8 bit	Esadecimale	EBCDIC 8 bit	Esadecimale	EBCDIC 8 bit	Esadecimale	EBCDIC 8 bit
0		2D		5A		88	h	B6		F3	T
1		2E		5B	$	89	i	B7		F4	U
2		2F		5C	*	8A		B8		F5	V
3		30		5D	)	8B		B9		F6	W
4		31		5E	;	8C		BA		F7	X
5		32		5F		8D		BB		F8	Y
6		33		60		8E		BC		F9	Z
7		34		61		8F		BD		FA	
8		35		62		90		BE		FB	
9		36		63		91	j	BF		FC	
A		37		64		92	k	C0		FD	
B		38		65		93	l	C1	A	FE	
C		39		66		94	m	C2	B	FF	
D		3A		67		95	n	C3	C	F0	0
E		3B		68		96	o	C4	D	F1	1
F		3C		69		97	p	C5	E	F2	2
10		3D		6A		98	q	C6	F	F3	3
11		3E		6B		99	r	C7	G	F4	4
12		3F		6C	%	9A		C8	H	F5	5
13		40	blank	6D	-	9B		C9	I	F6	6
14		41		6E		9C		CA		F7	7
15		42		6F	?	9D		CB		F8	8
16		43		70		9E		CC		F9	9
17		44		71		9F		CD		FA	
18		45		72		A0		CE		FB	
19		46		73		A1		CF		FC	
1A		47		74		A2	s	D0		FD	
1B		48		75		A3	t	D1	J	FE	
1C		49		76		A4	u	D2	K	FF	
1D		4A		77		A5	v	D3	L		
1E		4B	.	78		A6	w	D4	M		
1F		4C		79		A7	x	D5	N		
20		4D	(	7A	:	A8	y	D6	O		
21		4E	+	7B	#	A9	z	D7	P		
22		4F		7C	@	AA		D8	Q		
23		50	&	7D	'	AB		D9	R		
24		51		7E	=	AC		DA			
25		52		7F	"	AD		DB			
26		53		80		AE		DC			
27		54		81	a	AF		DD			
28		55		82	b	B0		DE			
29		56		83	c	B1		DF			
2A		57		84	d	B2		E0			
2B		58		85	e	B3		E1			
2C		59		86	f	B4		E2	S		
				87	g	B5					

Fig. 6-26: Tabella del codice EBCDIC

COMUNICAZIONE SINCRONA

Un formato di trasmissione asincrona contiene almeno due bit extra per carattere: start e stop. Quando i dati sono spediti con una sequenza continua di bit, senza start e stop, il ricevitore può perdere la sincronizzazione e modificare i dati in ingresso. Per prevenire questo, sono spediti *caratteri di sincronizzazione* ogni cento byte o più. Esiste la necessità logica, all'estremità ricevente, di risincronizzare la circuiteria di decodifica, con una frequenza sufficiente per non perdere la sincronizzazione. Usando tale metodo, conosciuto come *comunicazione sincrona,* sono presenti otto bit circa ogni 800 bit. Sono cioé presenti l'1% di dati in più rispetto al 20% del caso asincrono.

Sono state proposte varie forme di controllo di giunzione dati sincrona, («synchronous data link control» o SDLC). Alcune sono attualmente usate dalla IBM, dalla Burroughs, e da altri. Tutte hanno lo stesso formato base.

I dati sono trasmessi in blocchi di molti caratteri in un intervallo chiamato *trama.* Ciascuna trama ha un certo numero di *campi,* che contengono uno o più byte di dati. In figura 6-27 è indicata una trama di sette campi. Tutti i trasmettitori e ricevitori condividono la stessa linea. Soltanto uno alla volta può trasmettere e più possono ricevere. Ciascuna unità che vuole trasmettere deve aspettare finché la linea é occupata.

Il primo carattere indica al ricevitore che ha avuto inizio una trama.

Tutte le interfacce SDLC controllano l'inizio della trama. Se non c'é il loro indirizzo, ritornano in attesa. Se riconoscono il proprio indirizzo, ricevono l'intera trama e controllano eventuali errori. Una trama di risposta é inviata al trasmettitore per indicare se i dati sono stati ricevuti corretti o quanti errori sono stati riscontrati e se é necessaria la ritrasmissione.

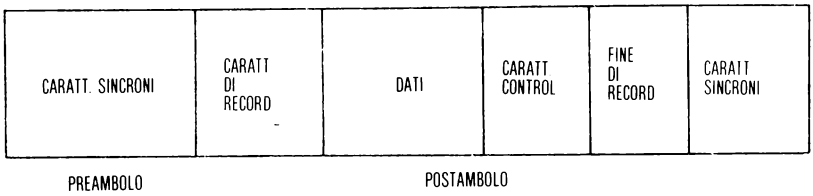


Fig. 6-27: Formato di dati sincroni

Possono essere trasmessi due o più tipi di trame. Esse differiscono nei byte di controllo che possono contenere informazioni differenti. Una *trama di informazione* contiene dati per il ricevitore indirizzato. Una *trama di protocollo* contiene dati di supervisione e gestione della rete di trasmissione. La parte dati della trama contiene un numero multiplo di byte di dati. Il formato dei dati può essere ASCII, binario o in altro codice.

I caratteri di controllo, CRCC o LRCC, sono usati per rivelare e correggere errori su un bit singolo e rivelare errori su due o più bit. I 32 bit usati contengono sufficiente informazione di controllo per far questo. Il carattere di stop é lo stesso del carattere di start e indica che la trama é terminata.

Nel processo di trasmissione, devono essere inseriti automaticamente caratteri o bit di risincronizzazione per mantenere la temporizzazione del sistema. Come nel caso delle etichette di indirizzo nel disco «floppy», i caratteri di start e di stop devono essere particolari configurazioni di 1 e 0 che sono facilmente riconosciuti da un'unità hardware.

I nuovi circuiti integrati adattatori seriali sincroni, come il controllore SDLC della Intel, il Motorola XSDA, o l'interfaccia seriale sincrona della Zilog, gestiscono i diversi protocolli attraverso una combinazione di sfotware del microprocessore e l'hardware del modulo di controllo. Un controllore tipico riconosce i caratteri di start, inserisce e cancella automaticamente bit di sincronismo, e svolge qualche controllo sul blocco. Il software deve comporre la trama con i campi presenti, e decodificare le trame attraverso l'esame di campi. Il programma ha anche il compito di sistemare il protocollo iniziale della rete.

RIVELAZIONE E CORREZIONE DI ERRORE

Molti componenti come le cassette, i dischi, le memorie dinamiche, i modem fanno errori nella scrittura e nella lettura dei dati. Esistono tre schemi fondamentali per rivelare tali errori: *la parità, il controllo di somma* e i *caratteri di ridondanza ciclica*. Per non rivelare soltanto ma anche correggere errori, sono necessarie altre informazioni. Sono per questo descritti due metodi di correzione di errore: i *codici di hamming* e la *parità trasversale.*

Parità

In un byte dati possono esserci un numero pari o dispari di 1. Il bit di parità é l'ottavo, o talvolta il nono bit, aggiunti ad ogni byte per rendere il numero di uni presenti pari o dispari. Gli errori possono essere rivelati registrando o memorizzando un bit di parità insieme ad ogni byte. A seguito della lettura del byte, risulta un dato valore di parità dagli otto bit letti. Se esso non corrisponde a quello memorizzato, esiste un errore su almeno un bit. *È da notare che i cambiamenti contemporanei di due bit, rispettivamente da 1 a 0 e da 0 a 1, non sono rivelati.*

Controllo di somma

Per verificare che un intero blocco di dati è corretto viene generato e aggiunto alla fine del blocco un byte di controllo. Per il controllo a seguito della lettura, il nuovo carattere costruito è confrontato con quello memorizzato alla fine del blocco. Se sono differenti nel blocco c'è un errore.

Il controllo di somma può essere generato facendo la somma di tutti i byte del

blocco mediante istruzioni di somma con riporto. Il numero di bit uno tra gli otto è pertanto «legato» al contenuto informativo del blocco. Altro modo è realizzare l'OR esclusivo tra tutti i byte. Il byte risultante è in realtà la parità dell'intero blocco, piuttosto che la parità del byte. Maggiore è la informazione del controllo di somma, più accurata è la rivelazione di errore.

Ridondanza ciclica

Essa è stata spiegata nei paragrafi relativi al disco «floppy» e pertanto il lettore faccia riferimento al capitolo quattro. Oltre all'algoritmo indicato per il formato dei dati del disco floppy possono essere usati altri algoritmi di controllo a ridondanza ciclica.

Codici di Hamming

Aggiungendo ridondanza al byte memorizzato, possono essere non soltato rivelati, ma anche corretti, errori su singolo bit.

Usando otto bit per il byte di dati, è possibile aggiungere ad esso un numero di bit pari a $(\log_2 8) + 1$ secondo la codifica di Hamming. Ciò implica l'uso di una parola di 12 bit per otto bit di dati. I quattro bit extra saranno bit di parità per sottogruppi differenti degli 8 bit originari.

b0 b3 b6 → h2
b1 b4 b7 → h3
b2 b5
↓ ↓
h0 h1 parità per righe
parità per colonne
b0b1b2b3b4b5b6b7 = byte
h0h1h2h3 = bit di Hamming.

Se la circuiteria di correzione trova che il bit generato h1 è differente dal corrispondente letto, e tutto il resto è corretto, è invertito il quinto bit. Se h1 e h2 sono errati, è invertito il terzo bit. Sono corretti tutti gli errori su singolo bit, mentre sono rivelati quelli su due bit.

Parità trasversale

Espandendo il concetto di Hamming ai blocchi può essere applicata la parità ai blocchi di byte, oltre che ai singoli byte. Può essere così costruito un vettore su ogni singolo bit errato.

Se i bit 10 e p0 sono errati, e invertito il bit b0 del primo byte. Questi schemi di rivelazione e correzione di errore sono stati semplificati nella descrizione fatta. Si presentano naturalmente problemi che richiedono ulteriore studio. Situazioni come 18 e p0 errati non permettono l'individuazione del bit errato in base a quanto indicato. Sono state svolte molte valutazioni sulle tecniche di correzione di errore per dati binari e il lettore è indirizzato ad esse per una piú completa descrizione dei criteri.

	b0		b0	10	parità attraverso
	b1		b1	11	il blocco
9		.....		.	
bit	.		.	.	
	b7	.	b7	.	
	p0	p1	pn	18	

BLOCCO
parità lungo
il blocco

STUDIO IN UN CASO: ECONOMICA PIASTRA ANALOGICA PER IL BUS S100

Il circuito in figura 6-28 mostra un convertitore da digitale ad analogico con capacità di conversione da analogico a digitale. Il circuito ha sei circuiti integrati: una porta nand tripla a tre ingressi, un decodificatore 74LS138, un circuito di pilotaggio del bus di tipo tri-state 74LS125, un controllore di trasferimento (latch) ottale 8212, un convertitore D/A MC1407, e un amplificatore operazionale quadruplo LM324. Mediante tali componenti è stato possibile progettare la struttura del bus S100 per misure analogiche.

Caratteristiche di tale modulo sono:

- Compatibilità al bus S100, soltanto 1 carico LSTTL per via del bus
- Risoluzione ad 8 bit per entrambe le conversioni D/A e A/D
- Conversione D/A in 20 μs
- Conversione A/D in 1 ms
- Da 0 a 10 volt in ingresso e in uscita con stadio di guadagno extra tra 1 e 1000 per ingressi a basso livello.

Il circuito sarà descritto in ogni sua parte, per indicare le funzioni di ciascun componente.

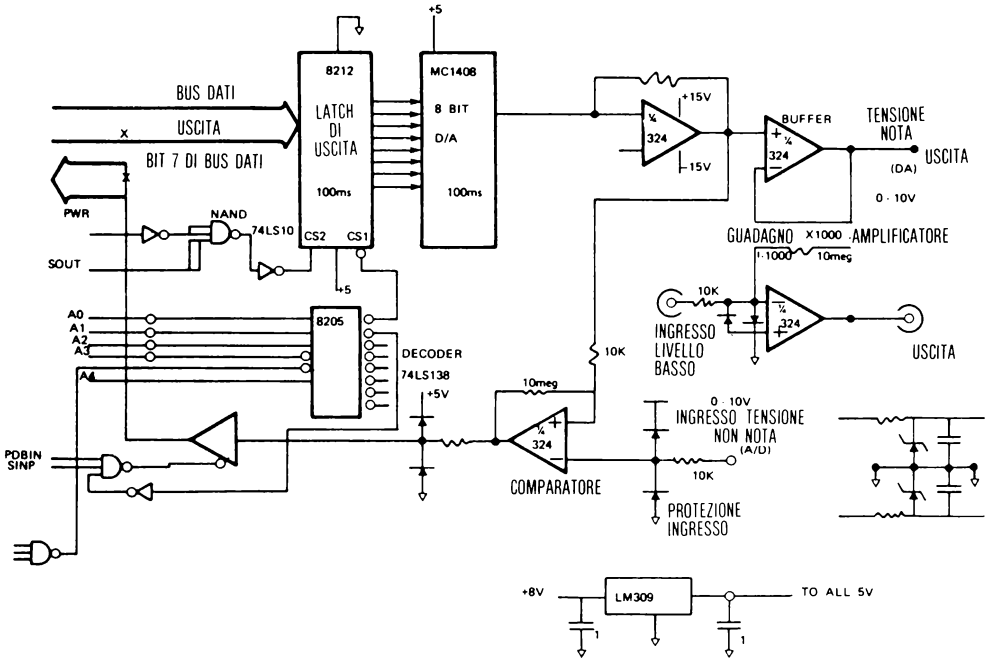


Fig. 6-28: Piastra A/D, D/A attraverso l'S100

L'hardware

Il bus dati di uscita che realizza tutti i trasferimenti di dati verso la memoria o le porte di uscita è connesso a un «latch» 8212. Ciascun bit è caricato da un ingresso del «latch». Ciascun ingresso rappresenta 2/3 del carico in ingresso di un «low-power - Schottky».

Il decodificatore 74LS138, insieme con i 74LS10 e 74LS04 decodifica l'uscita sulla porta «F8» (esadecimale). L'indirizzo è parzialmente decodificato da 1/3 del 74LS10 in modo che i bit A7, A6 e A5 siano tutti «1» per abilitare il decodificatore 74LS138. La prima uscita presenta «F0» negli otto bit di indirizzo a basso livello. Esso abilita uno dei selettori di circuito nel latch 8212.

L'altro selettore di circuito è pilotato dalla condizione $\overline{PRW}$ falsa e SOUT vera. Ciò è ottenuto invertendo $\overline{PRW}$ e ponendolo in NAND con SOUT. L'uscita del NAND è invertita verso il secondo selettore di circuito dell'8212.

In tal modo l'uscita del bus dati è controllata nel trasferimento verso l'8212 quando l'indirizzo è «F0», mentre il segnale di controllo indica che si sta eseguendo un'istruzione di uscita. La temporizzazione è indicata in figura 6-29.

Il dato, controllato nel trasferimento, è spedito al convertitore digitale-analogico MC1408. Una corrente proporzionale all'ingresso binario è presente all'uscita del convertitore. Per convertirla in una tensione è usato un circuito convertitore da corrente in tensione. Esso è realizzato con 1/4 dell'amplificatore operazionale quadruplo LM324.

L'uscita è ora una tensione tra 0 e 10 volt per un ingresso tra «00» e «FF» (esadecimale). Il secondo amplificatore operazionale, nello LM324, è usato per tamponare l'uscita, in modo da poter pilotare l'uscita, senza impegnare la sezione del comparatore.

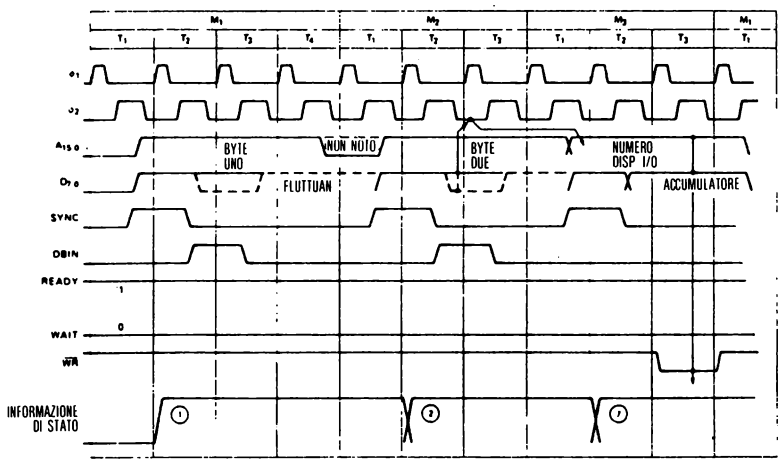


Fig. 6-29: Temporizzazione di un ciclo di scrittura in uscita attraverso l'S100

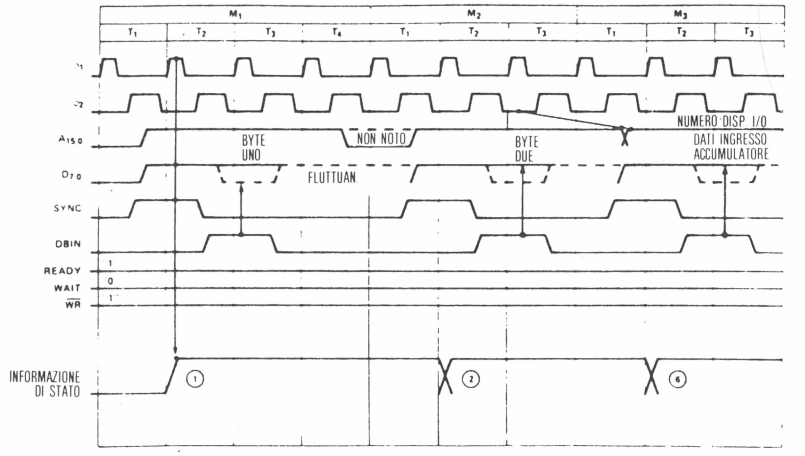


Fig. 6-30: Temporizzazione di un ciclo di lettura sul bus S100

Il terzo amplificatore operazionale è usato come comparatore per la conversione da analogico a digitale. L'amplificatore confronta l'ingresso non noto con l'uscita da D/A. Se l'uscita non nota è troppo piccola è usato il quarto A.0. per aumentare la tensione del segnale, mediante un amplificatore a guadagno variabile. Sono da notare i diodi di protezione, usati per evitare che siano prodotti guasti agli ingressi. I transitori sono infatti limitati a 100 volt.

L'uscita del comparatore è limitata ai livelli TTL dalla combinazione resistenza-diodo, in modo che può essere pilotato il 74LS125, circuito di pilotaggio a tre stati. Esso è abilitato da un comando di ingresso, e dall'indirizzo «F9» (esadecimale). La decodifica è fatta in modo simile alla porta di uscita, eccetto che la seconda uscita del 74LS138 è usata per decodificare l'indirizzo «F1». Inoltre le vie di controllo PDBIN e SINP sono poste in AND con l'indirizzo, per abilitare il circuito di pilotaggio del bit 7 del bus dati.

Pilotando il bit 7, si può realizzare un ingresso dalla porta «F1», ruotare il bit 7 sul bit di riporto, controllare tale riporto, per vedere se si è al di sopra o al di sotto della tensione non conosciuta. Presentando in uscita sulla porta «F0» un nuovo valore e controllando nuovamente «F7», si realizzano nell'insieme le funzioni basilare del convertitore. In figura 6-30 è indicata la temporizzazione per un'operazione di ingresso.

L'alimentazione è garantita da un regolatore di tensione a +5 volt per tutte le terminazioni Vcc, mentre i regolatori a diodo Zener forniscono la tensione a + e - 15 volt per gli amplificatori operazionali.

Si fa osservare che tre dei circuiti di pilotaggio del bus sono usati come invertitori. Come ciò è realizzato è indicato in figura 6-31.

Quando l'ingresso è a livello basso è abilitato il pilotaggio e l'uscita è portata su fino a un livello logico «1». Quando l'ingresso è invece alto, il pilotaggio è disabilitato e la resistenza di 240 ohm porta l'uscita ad un livello «0». Si sarebbe potuto usare un invertitore per realizzare tale funzione, ma si sarebbe aumentato il numero di componenti.

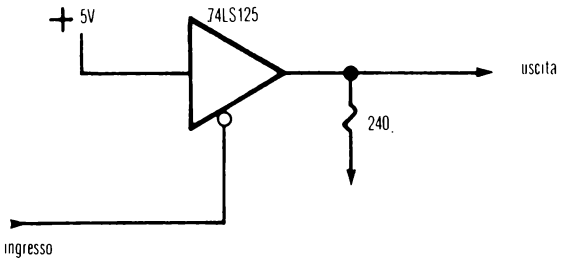


Fig. 6-31: Uso di un 74LS125 come invertitore

Il software

Per la conversione da analogico a digitale il valore binario che deve essere convertito è presentato in uscita sulla porta «F0». Ciascun salto di tensione, è pari a 10/256 = 0,0390625 volt. Ciò significa che se si vogliono 2,5 volt in uscita, il numero binario corrispondente è:

$$\frac{V\,out}{39{,}0625 \times 10^{-3}} = Num_{10} \xrightarrow[\text{in binario}]{\text{Convertito}} Bin_2$$

cioé:

$$\frac{2{,}5}{39{,}0625 \times 10^{-3}} = 64_{10} \longrightarrow 0100\,0000_2$$

o 40 in esadecimale. Naturalmente 80 corrisponde a 5 volt, poichè il convertitore è lineare. In software è necessario:

```
MOV A, M : poni il valore dalla memoria sulla uscita
OUT F0H  : uscita
```

DOMANDA: *Quale è la più alta frequenza alla quale si può lavorare con tale sistema?*

RISPOSTA: *Poichè il teorema di campionamento stabilisce che è necessario campionare, o, alternativamente, presentare in uscita un valore, almeno a velocità doppia della frequenza più alta, deve essere verificata la seguente relazione:*

$$\frac{1}{\text{velocità conversione}} \cdot 1/2 = f_{max}$$

cioé:

$$\frac{1}{20 \times 10^{-6}} \cdot 1/2 = 250 \text{ kHz}$$

In pratica il programma non è capace di prelevare informazioni a velocità tale da usare la banda indicata; si potranno produrre suoni di tipo musicale o vocale.

Conversione analogico - digitale

Per realizzare la conversione A/D è necessario realizzare in software l'algoritmo delle approssimazioni successive. Altra tecnica che può essere usata è quella della conversione a conteggio. Sono di seguito descritte entrambe.

Nel capitolo 5 è stata trattata l'approssimazione successiva. Per codificare questo metodo in una subroutine in linguaggio assembler dell'8080 è necessario esaminare il diagramma di flusso della figura 6-32.

In figura 6-33 è indicato un programma che realizza la conversione del diagramma nell'assembler. È da notare che questo programma usa le istruzioni «NOP» e «CMP E, M» per bilanciare la temporizzazione dell'istruzione «JC». In tal modo la conversione richiede lo stesso tempo per eseguire entrambe le diramazioni presenti nel diagramma.

Il tempo di conversione è 373,5 μ sec sulla base dei tempi di esecuzione delle istruzioni, senza tempi in attesa. È possibile pertanto campionare circa ogni 380 μ sec.

DOMANDA: *Quale è la più alta frequenza di campionamento?*

RISPOSTA: *nuovamente, in accordo al teorema del campionamento, essa è:*

$$\frac{1}{\text{tempo di conversione}} \cdot 1/2 = f_{max}$$

$$\frac{1}{380 \times 10^{-6}} \cdot 1/2 = 1316 \text{ Hz}$$

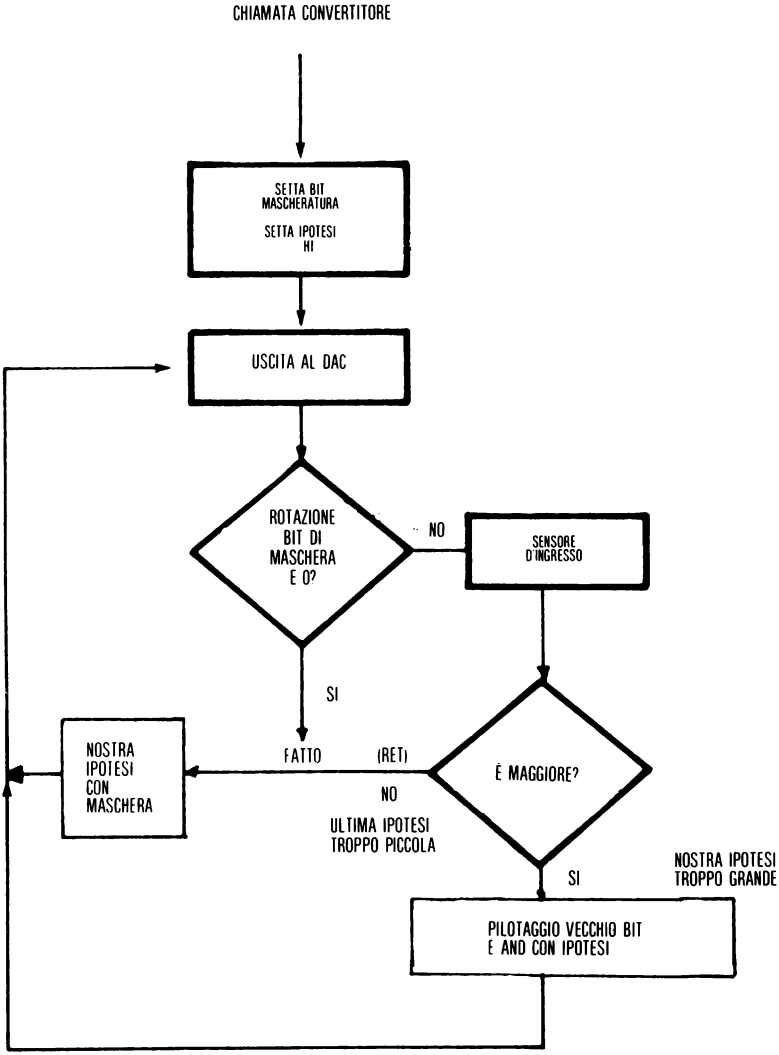


Fig. 6-32: Diagramma di flusso di approssimazione successiva

Ciò significa che il convertitore può lavorare ad una velocità appena sufficiente per rendere digitale la voce.

```
                MVI  D, 80H:     maschera di approssimazione in D
                MVI  B, 80H:     maschera in B
                MVI  C, 80H:     ipotesi in C
USCITA SUPPOSTA: MOV A,C

                OUT  DAC  :      IPOTESI IN USCITA

                MOV  A,B
                RRC
                RC               eseguito se bit riporto «settato»
                MOV  B,A
                IN   SENSE

                RLC
                JC   MAGGIORE
                MOV  A,D
                RRC
                MOV  D,A
                MOV  A,C
                ORA  B
                MOV  C,A
                CMP  E,M
                JMP  USCITA SUPPOSTA

MAGGIORE        MOV  A,D
                CMP
                AND  C
                MOV  A,D
                RRC
                MOV  D,A
                NOP
                JMP  USCITA SUPPOSTA
```

Fig. 6-33: Programma di una conversione A/D

ALIMENTATORI

Ora che il circuito è operativo, a che valore è la tensione a + 5 volt (± 5%) a 10 amperes? *Gli alimentatori*, che sono la base del sistema, sono spesso sottovalutati. Se l'alimentazione non ha specifiche ben definite, il sistema corre il rischio di non essere affidabile.

La tensione di alimentazione è caratterizzata semplicemente dai seguenti parametri:

- Livelli di tensione e di corrente
- Campo di regolazione
- Efficienza.

La descrizione del progetto di un alimentatore richiederebbe la scrittura di un altro libro di dimensione pari a questo. L'ingegnerizzazione degli alimentatori è una branca che è molto più difficile da imparare della programmazione e del progetto logico. Alla fine del capitolo sono indicati diversi buoni riferimenti sulla tecnica della alimentazione.

Livelli di tensione e corrente

Gli alimentatori sono limitati dal livello di energia o potenza che possono erogare al carico, attraverso le specifiche di tensione e corrente.

Per esempio un alimentatore di 5 volt e 5 ampere erogherà al massimo sul carico 25 watt.

Conoscere le tensioni che i circuiti richiedono comporta soltanto la lettura dei valori nei fogli delle specifiche. Ma quanta corrente assorbirà l'intero sistema? Nuovamente, le esigenze di corrente sono indicate nelle specifiche dei componenti. Occorre sommare tra loro i valori indicati nel loro valore tipico, medio, e massimo, o alternativamente sommare la dissipazione di potenza indicata. Si ottiene così un valore approssimato della corrente di lavoro. Per fissare le specifiche dell'alimentatore è necessario prevedere un livello di corrente doppio del valor medio o tipico richiesto. Devono essere indicati anche i valori massimi richiesti, così come il campo di regolazione minimo dell'alimentatore.

Regolazione

Gli alimentatori non sono perfetti. Essi non possono fornire 5 volt per tutte le condizioni di carico. Questo è il motivo per il quale essi devono essere caratterizzati anche nel loro *campo di regolazione*, o nella capacità di mantenere la tensione di uscita costante.

Le specifiche sono suddivise nella *tolleranza di regolazione a carico*, e nella tolleranza di *regolazione a vuoto e a carico*. Sono anche importanti l'*impulso di sovratensione all'accensione*, e la *stabilità* sotto diverse condizioni di carico.

Se il carico è costante e la tensione di linea di ingresso può assumere qualunque valore nel campo garantito dalla rete, la variazione della tensione di uscita al varia-

re della temperatura e del tempo è *la tolleranza di regolazione a carico.*

Per esempio, se il sistema è un microcalcolatore 8048 in singolo modulo, è necessaria una regolazione a carico del ± 5% : il campo di operazione garantito è tra 4,75 e 5,25 volt. L'alimentatore deve presentare caratteristiche tipicamente due volte superiori a quelle di sicurezza.

Si lavorerebbe altrimenti al limite del campo garantito. Combinando le tolleranze con altri fattori marginali del circuito, quali il carico dei bus, la temperatura e la frequenza del clock, il sistema potrebbe non funzionare.

Se il carico è variabile risulta importante conoscere le variazioni della tensione. Esse sono indicate con il termine misure della *regolazione a vuoto e a carico*, o *misure a commutazione di carico.* Ad esempio se si dispone di un alimentatore a 5 volt e a 5 ampere, l'alimentatore lavora con i valori nominali se è caricato con un ohm. Se è disconnesso il carico l'alimentatore dovrebbe teoricamente *non modificare* le sue caratteristiche. In realtà, una tolleranza minore dello 0,5 per cento è normale.

Oltre alla tolleranza sono molto importanti le misure dell'impulso di sovratensione e la stabilità. La *sovratensione* si può presentare quando l'alimentatore è acceso o spento. È importante conoscere il suo livello.

La *stabilità* indica l'eventuale presenza o meno di oscillazioni sotto condizioni di carico variabile.

Se infatti sono pilotati circuiti TTL standard, essi non tollerano un impulso di sovratensione superiore a 8 volt, altrimenti il circuito si distrugge. Gran parte degli alimentatori commerciali non presentano impulso di sovratensione o esso è di limitata entità.

In contrasto, molti alimentatori progettati in casa tendono ad essere instabili. Ciò è dovuto allo schema di interconnessione e alle tecniche costruttive. Occorre infatti ricordare che anche se la massima frequenza teoricamente presente è quella di rete, si possono innescare oscillazioni a frequenze nel campo del megaherz. Come mai? Il regolatore deve rispondere velocemente alle variazioni di carico per rispettare le specifiche indicate. Più rapida è la risposta alle regolazioni e alle variazioni di carico, più velocemente il circuito deve autoregolarsi. Ciò può essere causa di instabilità se l'alimentatore non è correttamente progettato.

Efficienza

Se l'alimentatore eroga 25 watt al carico, quanti watt assorbe dalla rete? L'efficienza è il rapporto tra la potenza di uscita e quella di ingresso. Alimentatori tipici presentano una efficienza del 40%. Cioè per l'alimentatore a 25 watt indicato saranno assorbiti dalla rete 62,5 watt. Esistono regolatori di tipo *switching* che presentano una efficienza fino al 90%. Essi sono però più costosi dei regolatori lineari.

Come scegliere l'alimentatore

La domanda che ci si pone sempre è: «lo compro o lo sviluppo?». Sarà prima esaminato l'acquisto. I tipi migliori sono quelli conosciuti come O.E.M. o «Original

Equipment Manifacturer». Essi sono unità commerciali che sono usate nei prodotti di molte società. Essi sono costosi, tipicamente costano 50 $ per un alimentatore di 35 watt, o 5 volt a 7 ampere. Vantaggio di un alimentatore O.E.M. è che l'utente sfrutta centinaia di migliaia di dollari investiti nella ingegnerizzazione degli alimentatori. Regola empirica è che il suo costo è di circa 1,5 $ per ogni watt erogato.

Se si progetta l'alimentatore, occorre fare molte scelte di progetto. I trasformatori, i diodi, le capacità devono essere scelti in base ad alcune regole e formule di progetto. Il regolatore stesso deve poi essere ben accoppiato alla combinazione del trasformatore, dei diodi, e del condensatore in modo che non si presentino problemi di stabilità ed efficienza.

Per applicazioni con meno di 3 ampere a 5 volt sono disponibili semplici regolatori di tensione monolitici, come il popolare LM309 o quelli della serie 78XX. Gli schemi di specifica di tali regolatori contengono tipiche strutture circuitali per le loro diverse applicazioni. Al di sotto di 15 vatt si suggerisce la lettura dei riferimenti bibliografici indicati. È importante ricordare che l'alimentatore deve poter lavorare coprendo le variazioni delle caratteristiche di assorbimento dei carichi ed è buona norma non fermarsi all'esame del comportamento del solo prototipo progettato.

La figura 6-34 indica le caratteristiche tipiche di un alimentatore OEM prodotto dalla Power - One Corporation. In figura 6-35 è rappresentata la sua struttura.

Ingresso AC:	105-125 VAC, 47-440 Hz (Variazione di corrente di uscita del 10% per funzionamento a 50 Hz.)
Uscita DC:	Vedi campo di regolazione del diagramma limiti di tensione/corrente, ± 5% al minimo
Regolazione di linea:	± 0,01% per variazioni di linea del 10%
Regolazione di carico:	± 0,02% per cambiamenti di carico del 50%
Ondulazione di uscita:	1,5 mV picco-picco, 0,4 mV massimo Val. Eff.
Risposta transitoria:	30 sec per variazione del 50% del carico
Protezione Corto C. e di sovraccarico:	Limitazione/autoriduzione autom. di corrente
Protezione per tensione inversa:	Presente in uscita
Lettura remota:	Presente, se abilitata e presente protezione
Stabilità:	± 0,05% per 24 ore dopo riscaldamento
Limiti di temperatura:	0°C ÷ 50°C con dati di targa, riduzione lineare al 40% a 70°C
Coefficiente temperatura:	± 0,01%/°C al massimo, 0,002% tipico
Efficienza:	Unità a 5V:45%; unità a 12 e 15V: 55%; unità a 20 e 24V: 60%
Vibrazioni:	Su base Mil-Std-810B, metodo 514, procedura 1, curva AB (a 50 Hz)
Shock:	Su base Mil-Std-810B, metodo 516, procedura V

Fig. 6-34: Specifiche di un alimentatore

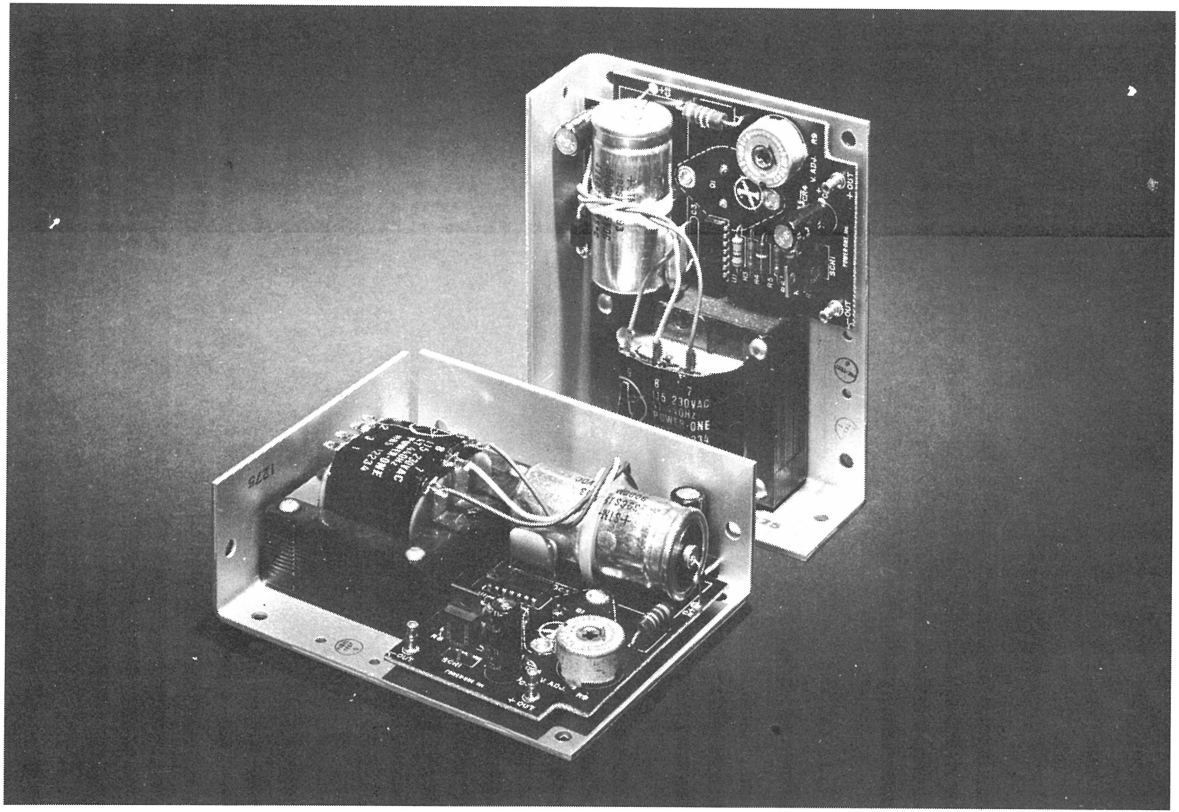


Fig. 6-35: Alimentatore OEM

CONCLUSIONI

È stata progettata una piastra analogica di raccolta e di controllo. È stata progettata per essere connessa al bus S100. È stato scritto il software per utilizzare le caratteristiche di tale convertitore D/A e A/D.

Gli standard e i bus descritti hanno lo scopo di interfacciare più semplicemente. Sulla base della considerazione che connettere il componente semplicemente al sistema, senza lavoro extra, è puramente un sogno, si è visto che molti utilizzatori degli standard S100, CAMAC, IEEE-488 e EIA-RS232C creano la necessità di componenti, moduli e sistemi compatibili agli standard. È pertanto preferibile *restare entro gli standard.* Il progetto risulta più semplice e il tempo può essere impiegato per problemi più complessi.

Sono stati descritti esempi di standard di bus seriali e paralleli, metodi di comunicazione tra moduli, e una interfaccia reale verso il bus. Il bus S100 è attualmente il bus parallelo più popolare; sono infatti state prodotte più di 600 tipi di piastre diverse compatibili con esso. Lo standard seriale RS232C è quello più popolare per la comunicazione dati. Sue applicazioni sono per la memorizzazione e il prelievo di dati da cassette via modem, mediante opportune tecniche di formato. Tali tecniche sono descritte in dettaglio nel capitolo 4.

Gli alimentatori sono alla base dei sistemi. Sono stati discussi la regolazione, la stabilità e alcuni parametri di progetto. La soluzione OEM è quella proposta per la utilizzazione della esperienza di specialisti nel progetto.

CAPITOLO 7

IL MULTIPLATORE STUDIO DI UN CASO

INTRODUZIONE

Questo sistema ha lo scopo di concentrare 32 terminali compatibili con l'interfaccia EIA RS232C in una linea di trasmissione contemporanea in entrambe le direzioni ad alta velocità. Ciascun terminale dispone di una memoria temporanea in uscita e un ingresso carattere per carattere. Pertanto l'elaboratore connesso ai terminali può ridurre i suoi tempi di servizio eseguendo la multiplazione dei canali.

Progettato per un PDP 11/70, il sistema é anche applicabile, soltanto con cambiamenti nel codice dell'elaboratore, a qualunque altra macchina. Il costo per fornire questa funzione é soltanto di $ 50 per canale, a confronto con il costo comune di circa $ 250 per canale. Il sistema é anche conveniente per raggruppamenti di meno di 32 terminali.

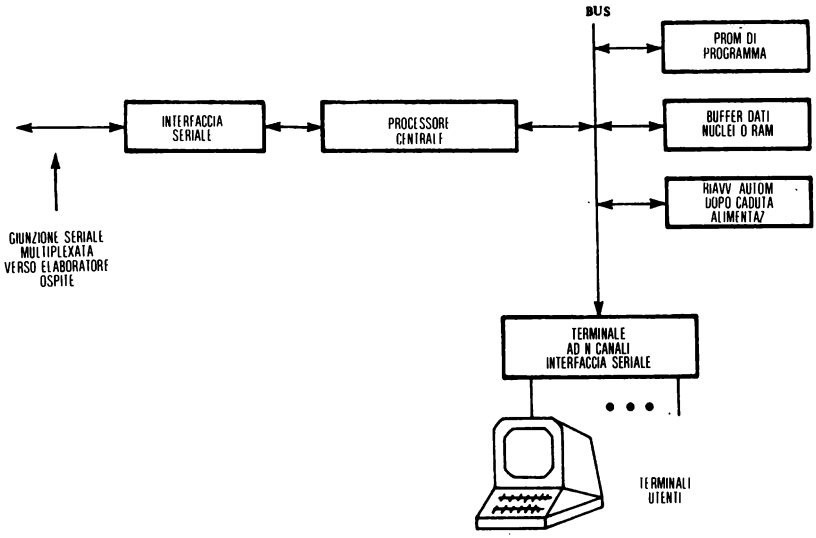


Fig. 7-0: Multiplatore a 32 canali

Il sistema usa il microprocessore 8080, l'USRT 8251, il controllore di interruzione 8259 e altri componenti della famiglia 8080. Il sistema non ha caratteristiche di controllo di modem poiché esso é stato impostato per essere nello stesso luogo dei terminali, risparmiando in costi di manutenzione e di cavi per connessione. Il costo indicato non comprende l'economia risultante dall'uso di un minor numero di linee telefoniche e di modem.

LE SPECIFICHE

L'abilità di connettere un gran numero di terminali a un servizio in «time-sharing» presenta sempre un gran numero di problemi di ingegnerizzazione. Molto deve essere fatto per i guai di interconnessione con i modem, del cablaggio telefonico, dell'adattamento di pannelli per test, e dell'interfacciamento interno della macchina.

Concentratori remoti avrebbero eliminati molti problemi. È da considerare, però, un nuovo problema: il costo. Lo scopo del progetto é in tal caso di servire 32 terminali ad una velocità di ingresso che non supera mai 30 caratteri al secondo, e una velocità di uscita quanto più possibile elevata. Poiché l'8080A può eseguire approssimativamente 300 istruzioni tra la gestione di caratteri su linea a 9600 baud, se fossero da servire 32 terminali in uscita sarebbero disponibili meno di 300 istruzioni per la gestione della lista di polling dei terminali. Ogni intervallo di tempo lasciato disponibile sarebbe usato per l'uscita. Il codice dovrebbe essere esaminato byte per byte, con una ottimizzazione del codice attentamente valutata. È stato sviluppato un prototipo, con l'obiettivo di servire almeno 16 terminali in modo graduato.

La statistica tipica dei nostri ingressi era un massimo di 150 baud al secondo, e una velocità complessiva di 50 baud per tutti i 32 terminali. Pertanto, una volta sviluppato, il multiplatore avrebbe potuto gestire al massimo 150 baud alla volta per tutti i canali, o un massimo di 300 baud per uno singolo. L'uscita sarebbe stata un minimo di 300 baud complessivi per tutti i canali, e un valore tipico di 600 baud, quando fosse fatta richiesta di servizio per un singolo terminale.

ARCHITETTURA

In figura 7-1 é indicato lo schema a blocchi della architettura. Ciascun terminale ha il suo proprio USART, poiché ciascuno richiede una interfaccia seriale dedicata. Gli USART sono raggruppati in quattro, e quindi disposti su piastre che sono connesse al bus del sistema 8080. Sono presenti 8192 bytes di RAM per memorizzazione di dati e 1024 bytes di EAROM per il programma, nel sistema. Infine, é presente un controllore di interruzioni e una piastra per il canale da alta velocità, che é connessa al bus.

Ciascun terminale, mediante il suo USART, ha associato a sé una memoria temporanea di 128 caratteri, per controllare il trasferimento in uscita verso il terminale.

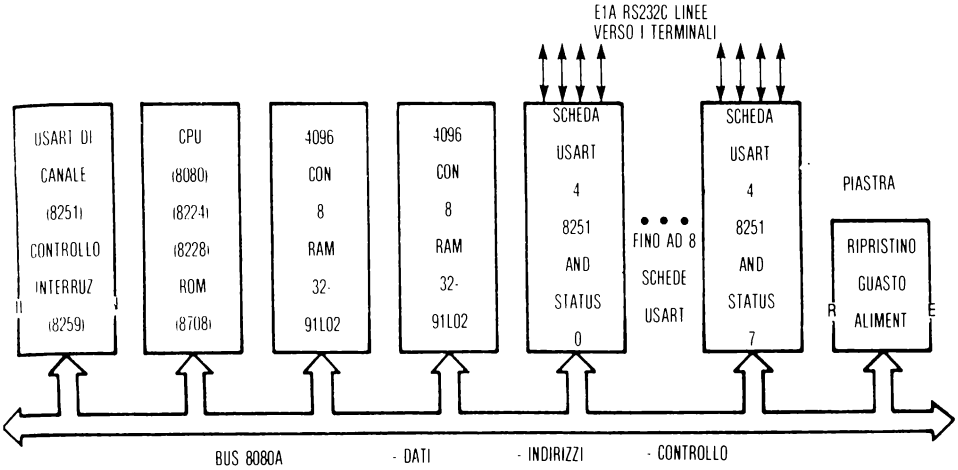


Fig. 7-1: Diagramma a blocchi del multiplatore

Ciò occupa 4096 bytes della RAM disponibile. La coda dal terminale verso l'elaboratore é lunga 256 caratteri. Tali lunghezze sono state scelte per ottimizzare i trasferimenti sul canale di trasferimento. Il metodo usato non é qui descritto.

Ci sono tre processi in fase di elaborazione, uno alla volta: «routine» di «polling» per servizio ingresso-uscita, processo di interruzione per la memoria tampone dall'elaboratore verso il terminale, e il processo di interruzione per la coda del terminale verso l'elaboratore.

SOFTWARE

Nelle figure 7-2 7-3 7-4 e 7-5 sono indicati diagrammi di flusso del software.

Il software può essere diviso in quattro parti la routine di inizializzazione, la routine di polling, la routine di interruzione per caricare le memorie tamponi con i dati dell'elaboratore e la routine di interruzione per riempire la coda d'attesa del terminale verso l'elaboratore.

L'inizializzazione è elaborata solo in fase di «reset». Successivamente possono essere elaborati gli altri processi, uno alla volta. Esse comunicano soltanto attraverso le memorie tamponi per uscita dati e non condividono nessun altro spazio di memoria comune, oltre alle tabelle del puntatore.

La routine di inizializzazione azzera tutta la memoria, sotto le tabelle, individua quali piastre sono inserite, resetta tutte le USART, e può stampare indicazioni di errore, se è installata la piastra di «debug». Questa è in sostanza l'intera struttura del sistema. Esso setta il puntatore dello stack, resetta e setta i modi, la velocità e il numero di bit per parola nella USART.

La sezione del programma copre il 60% del codice usato per l'intera applicazione.

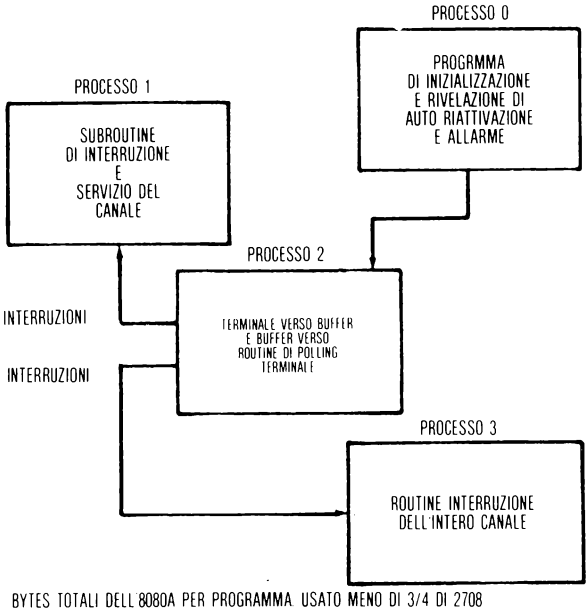


Fig. 7-2: Software del multiplatore: flusso dell'intero programma

La routine di polling procede lungo la lista definita dal programma di inizializzazione, verificando se é stato scritto un carattere da un terminale, o se é presente un dato in una memoria temporanea da presentare in uscita verso un terminale. Pertanto ciascun terminale é servito una sola volta durante una scansione della lista. Se il canale verso l'elaboratore é occupato (é necessario un millisecondo per trasferire un carattere a 9600 baud), i caratteri sono posti in una coda di attesa che sarà servita quando si verifica l'interruzione «canale-non-impegnato». Se il canale non é impegnato la coda di attesa é svuotata di un carattere e il carattere che é attualmente in attesa é posto nella coda sulla terminazione di linea. In tal modo la routine di servizio di coda é attivata e continua a produrre interruzioni, quando non in esecuzione, per svuotare tutti i caratteri in attesa del canale. Per la trasmissione dei dati é utilizzato il seguente formato: é inizialmente spedita al terminale la indicazione di avvenuta trasmissione, successivamente é spedito il carattere all'elaboratore con la routine di coda. Ciascuna piastra ha la propria tabella di priorità, in modo che é elaborato soltanto un ingresso, per passo e per piastra. Dopo che é stato trasmesso un carattere o, se la piastra registrata non ha caratteri, é verificata l'area di memoria temporanea di ciascun terminale per vedere se esiste un carattere in uscita disponibile. Tali caratteri sono posti nella memoria temporanea dalla routine di interruzione di elaboratore. Se la verifica dà esito positivo, il carattere dalla memoria temporanea é caricato sull'USART, per essere trasmesso, e sono aggiornati tutti i

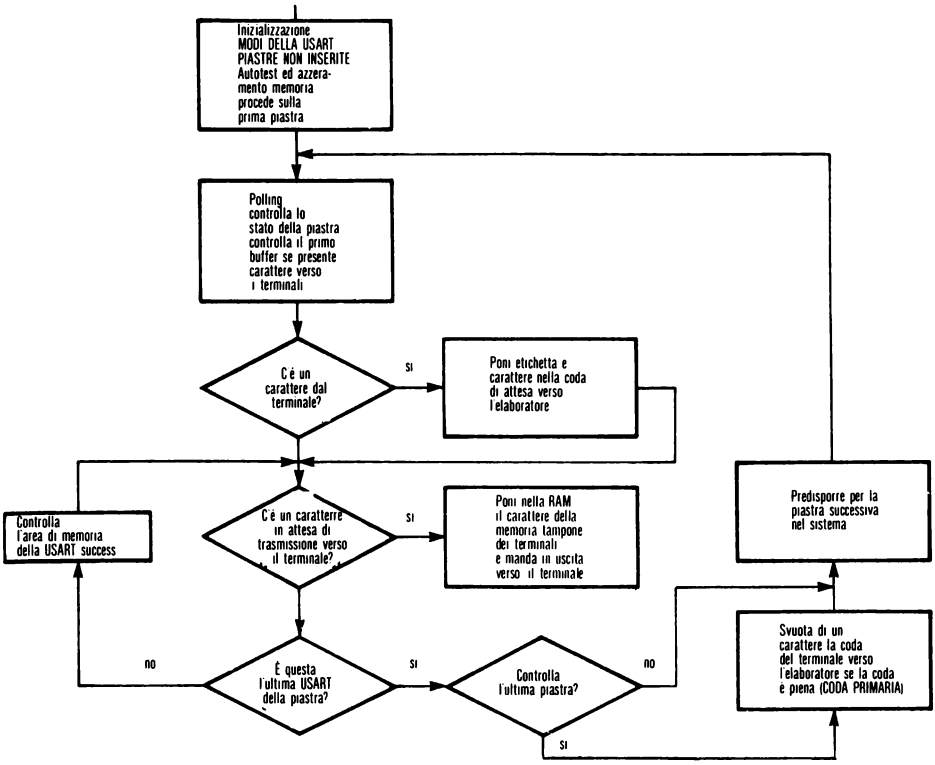


Fig. 7-3: Software del multiplatore: anello di polling

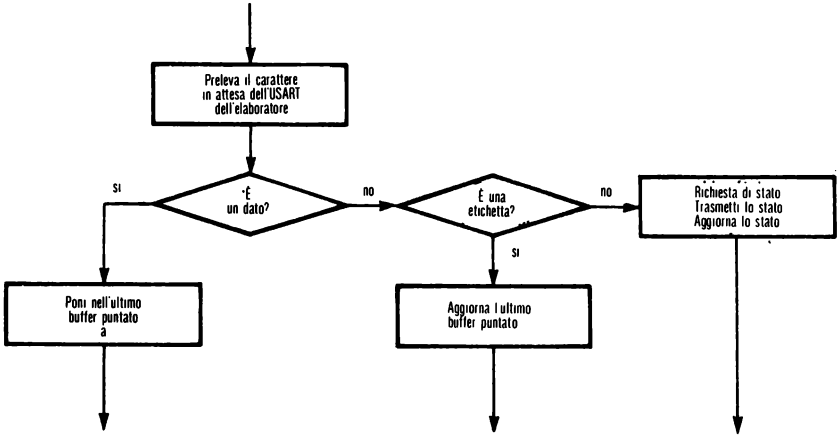


Fig. 7-4: Software del multiplatore: interruzione dall'elaboratore verso il multiplatore

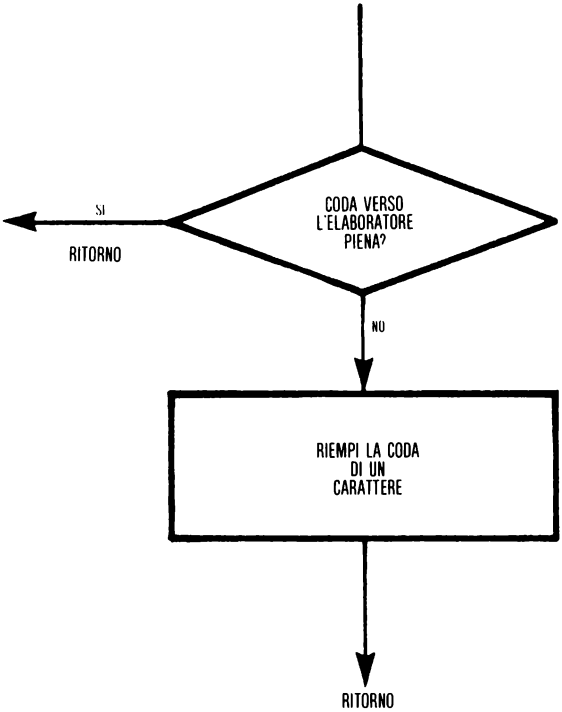


Fig. 7-5: Software del multiplatore: interruzione della coda del multiplatore verso l'elaboratore

puntatori. Quando non ci sono caratteri verso il terminale e nessuna memoria temporanea é piena, il sistema continua a registrare ciascuna piastra per l'ingresso, e ciascuna memoria temporanea dell'UART per l'uscita.

La routine di interruzione di coda di canale controlla le code, e trasmette un carattere se ne esiste uno in attesa; altrimenti termina. La routine non sarà richiamata nuovamente mediante interruzione, finché la routine di polling la attiva spedendo un carattere.

La routine di interruzione di elaboratore attende l'arrivo dell'informazione dal PDP 11/70, prima di essere eseguita. Quando é ricevuto un carattere e la routine é libera, é generata una interruzione, che dà inizio al processo relativo. Esso verifica il carattere in ingresso e, se esso é un dato, lo pone nella area di memoria temporanea di uscita appropriata. Dopo ciò, riprende il polling. Altri tipi di carattere dall'elaboratore comportano richieste di stato, commutazione di appendici ai dati e comandi di ri-inizializzazione software.

La routine di interruzione dell'elaboratore può comportare la interrruzione in qualunque momento durante il polling. Essa inizialmente salva il vettore di stato della macchina, successivamente tratta il carattere che ha causato l'interruzione. Se

il bit più significativo (MSB) é un «1», il carattere é un'appendice ai dati o un comando. Se esso é una appendice é memorizzato, in modo che i caratteri dati successivi siano caricati nella memoria temporanea alla quale ha puntato l'ultima appendice ai dati.

Il bit più significativo può anche indicare che il carattere é uno dei seguenti comandi: «richiesta di stato», «cambiamento di stato», e «ri-inizializzazione software». Una richiesta di stato comporta la spedizione nella direzione opposta di un'appendice di stato seguita dallo stato dell'USART in questione. Un cambiamento di stato comporta il prelievo del carattere successivo e il suo trasferimento al registro di controllo dell'USART. Esso può essere usato per porre «porte» in stato «on» o «off» e per cambiare la velocità in baud di un fattore quattro. La ri-inizializzazione software ri-inizializza l'intero sistema. Deve essere posta attenzione nell'uso di tali controlli: tali comandi richiedono un tempo maggiore di quello necessario per registrare tutti i terminali. Le memorie potrebbero non essere sufficienti a gestire il flusso dati correttamente perché alcune interruzioni non sarebbero servite, con conseguente perdita di dati. Tali comandi sono normalmente usati per riinizializzare il sistema da parte dell'elaboratore, dopo un malfunzionamento di quest'ultimo.

Il valore «0» del bit più significativo indica che il carattere é un dato. Tale carattere é pertanto caricato nell'ultima posizione della memoria tampone indicata come ultima appendice. Tutti i caratteri successivi sono caricati nella stessa memoria temporanea, finché é spedita una nuova appendice.

Il modulo CPU e PROM

In figura 7-6 é rappresentato lo schema della piastra CPU 8080. Essa contiene tutta la circuiteria di interfaccia di CPU necessaria insieme alla ROM programmabile 2708 ed alle memorie temporanee di bus necessarie.

Lo 8080 richiede un segnale di temporizzazione e un controllore di sistema. Tali funzioni sono realizzate dai moduli 8224 e 8228, rispettivamente. Lo 8224 fornisce le temporizzazioni necessarie dal cristallo a 18 megaherz per produrre il clock a due fasi dell'8080. Esso inoltre fornisce il segnale di reset necessario per la sincronizzazione. Il controllore di sistema 8228 fornisce al sistema il bus di controllo oltre a tamponare i dati del bus relativo, in modo che tutti i moduli del sistema siano pilotati senza limitazioni di carico.

In tale piastra sono anche presenti 1024 byte di EPROM nella 2708. È da osservare che la selezione di tale modulo é totalmente decodificata. La EPROM risponde solo a indirizzi da «0000» esadecimale a «03FF» esadecimale. In tale campo risiede il programma di multiplazione.

La selezione é realizzata nel modo di seguito descritto: tutti i bit di indirizzo da A10 fino ad A15 devono essere bassi per abilitare la EPROM, allo stesso modo come il segnale $\overline{\text{MEMR}}$. I primi quattro di tali segnali, insieme a $\overline{\text{MEMR}}$, vanno a un decodificatore da 1 ad 8, un 8205. Se essi sono tutti a zero, allora é selezionata la prima uscita. Poi questa uscita é controllata con le ultime due vie di indirizzo. Se

tutte sono a zero, allora $\overline{CS}$ é posto a livello basso, selezionando la EPROM. Il circuito di pilotaggio del bus della EPROM, un 8212, é anche a tal punto abilitato per pilotare la appropriata cella di dati nel bus dei dati, che sarà letto dal processore.

Moduli RAM

Nel sistema considerato sono presenti due piastre di memoria. Esse sono eguali tranne che nell'indirizzo. Mentre una copre il campo di indirizzi esadecimali tra «1000» e «1FFF», l'altra copre il campo «2000» – «2FFF». La memoria RAM disponibile è di 8192 bytes.

Ciascuna scheda contiene 32 circuiti integrati RAM statici di 1024 x 1 bit, circuiti di pilotaggio e di ricezione dal bus e logica di selezione indirizzo.

Un singolo modulo RAM può memorizzare 1024 bit di informazione. Per memorizzare 4096 x 8 bit é necessario organizzare tali moduli secondo una *schiera di*

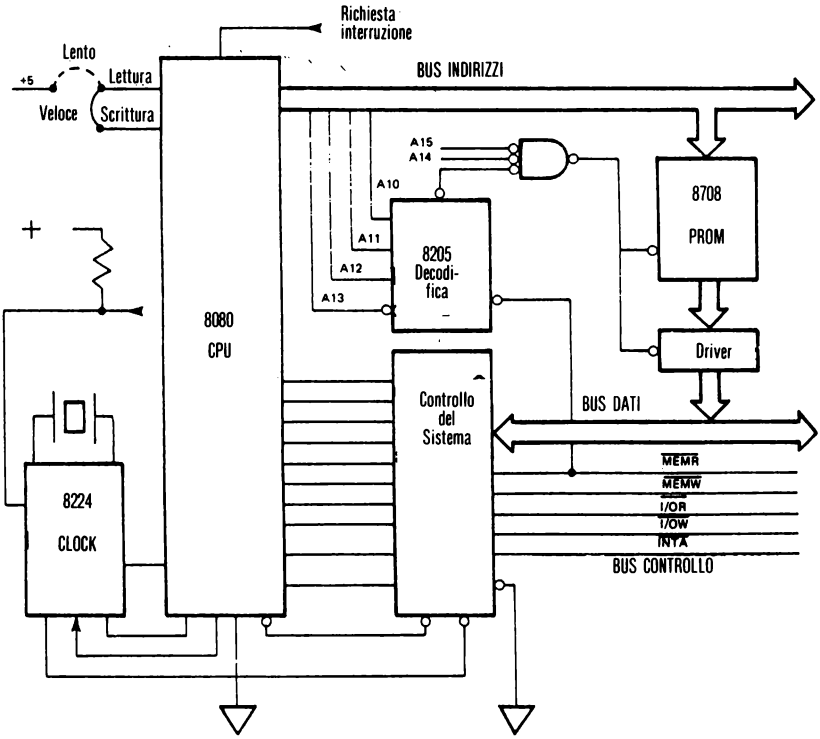


Fig. 7-6: Schema di piastra CPU

memorie. È da osservare che é necessario un modulo per ciascun bit e che é necessario utilizzare 4 gruppi di 4096 byte.

Poichè devono essere abilitati otto 91L02 per ciascun gruppo di 1024 byte, i selettori di modulo di ciascuno dei gruppi contenenti gli otto bit di byte sono interconnessi. Da qui, quello tra i 4 gruppi selezionato va al decodificatore uno di otto.

I bit dati sono da ciascun gruppo trasferiti attraverso un bus fisicamente perpendicolare al selettore di modulo. Tutti i bit di pari ordine, primo, secondo, terzo, etc., sono interconnessi. Poiché i 91L02 non possono pilotare il bus direttamente, tutte le vie di dati di ingresso sono caricate sul bus da un modulo di ricezione e pilotaggio bidirezionale 8216. In modo simile, tutti i dati in uscita dal 91L02 vanno ad un modulo di pilotaggio di bus bidirezionale. In figura 7-9 b é indicata una illustrazione dell'8216.

Due di questi componenti forniscono un metodo standard di predisposizione a

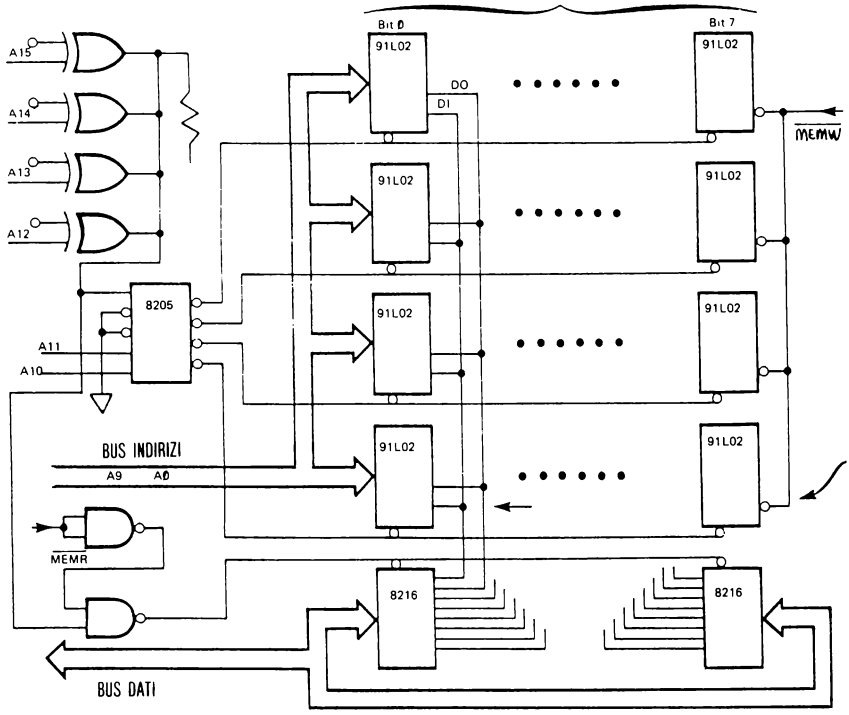


Fig. 7-7: Piastra RAM

ricevere e pilotare il bus dei dati. Il segnale $\overline{DIEN}$ controlla se il bus é pilotato dall'8216, o se il modulo é predisposto alla ricezione. Il segnale $\overline{CS}$ abilita le uscite per pilotare sia il bus che le uscite DO. Se $\overline{CS}$ é alto, tutte le terminazioni DB e DO sono nello stato di alta impendenza.

La direzione del flusso dati é determinata dal segnale $\overline{MEMR}$. Quando esso é basso, la RAM pone i dati in uscita sulle vie DO dell'8216. Sono abilitati i circuiti di pilotaggio del bus per fornire i dati al bus-dati dell'8080. In tutti gli altri intervalli la schiera di memoria é predisposta a ricevere dal bus. L'unico intervallo di tempo durante il quale il processore scrive i dati nella memoria è quando il segnale $\overline{MEMW}$ commuta al livello basso, e i moduli sono selezionati.

La selezione degli indirizzi é realizzata in modo che gli indirizzi della piastra possano essere selezionati mediante connessioni con ponticelli. I dieci bit di indirizzo più basso vanno direttamente ai 91L02.

Gli altri due bit vanno a un decodificatore 1 su 8 (8205), per selezionare uno dei quattro gruppi di otto circuiti integrati di memoria. La via di abilitazione dell'8205 risulta da una combinazione connessa in AND di porte or-esclusivo (XOR).

Soltanto quando tutte le uscite da queste quattro porte sono alte sarà abilitata la piastra di memoria. Ciascuna porta XOR confronta un bit di indirizzo con un ponticello posizionato a uno o a zero. Quando sono identici, l'uscita sarà «0». Quando

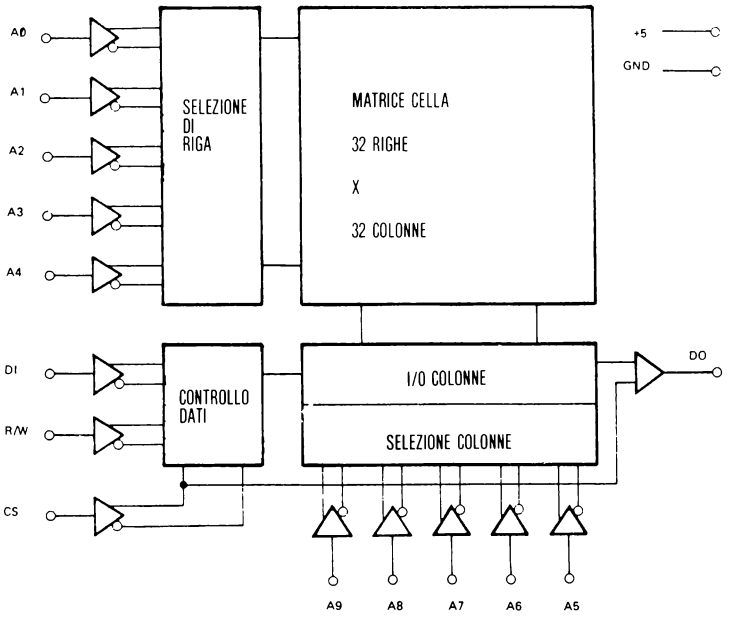


Fig. 7-8: Dettagli del 91L02C

sono differenti, l'uscita sarà «1». Per posizionare i ponticelli all'indirizzo giusto, occorre assegnare il valore negato dell'indirizzo che si vuole selezionare. Se si vuole «0010», per A15 - A12, i ponticelli sono connessi a «1», «1», «0», «1». In tal modo la piastra risponde soltanto se l'indirizzo é entro l'area $0100XXXXXXXXXXX_2$. Esso corrisponde alle pagine esadecimali tra «20» e «2F» esadecimali, o «2000» e «2FFF». *Ci si eserciti ad essere un lettore pronto: come sarebbero stati posizionati i ponticelli per il campo «1000» - «1FFF»?*

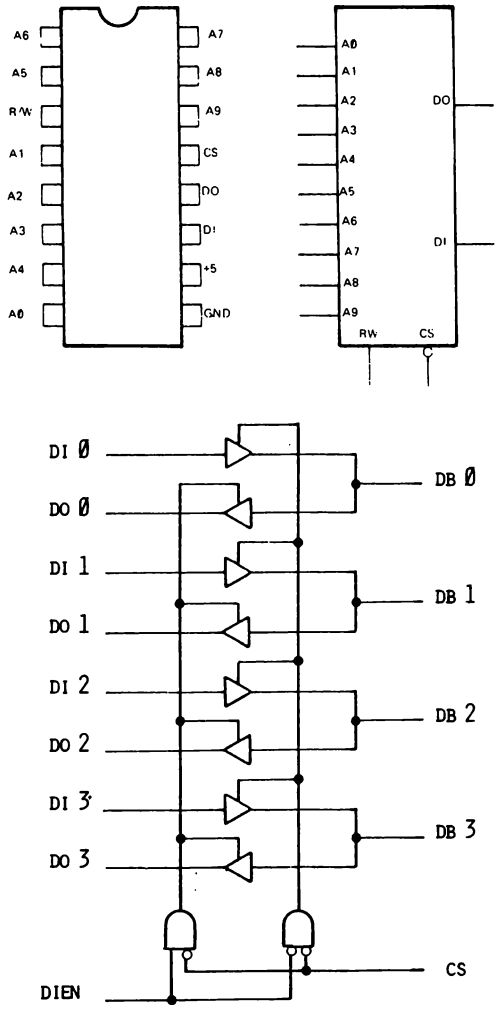


Fig. 7-9: Terminazioni del 91L02C (in alto) e Pilotaggi bidirezionali di bus 8216 (in basso)

La piastra USART

In figura 7-10 é indicata la piastra base per l'interfaccia di tutti i terminali. Questa piastra contiene quattro USART 8251, un generatore di clock per velocità in baud, e una PROM per la generazione della codifica delle priorità.

L'8251 é l'elemento fondamentale per l'interfaccia seriale. Presenti in gruppi di quattro per piastra, essi sono interconnessi nei loro bus per dati in modo da formare un unico bus dati tra le diverse piastre.

In modo simile alla piastra di memoria, questo bus é interfacciato dalla memoria tampone 8216 sul bus di sistema. Ciò perché l'8251 non può pilotare più di otto altri carichi LSTTL. L'8251 é selezionato mediante la tecnica di decodifica di indirizzo, usando un 8205. È da notare che questi componenti sono ingressi-uscite mappati in memoria.

Cioé, poiché gli stessi segnali che controllano la memoria ($\overline{MEMW}$, $\overline{MEMR}$) controllano anche gli USART, essi appaiono come posizioni di memoria. In accordo alla mappa in memoria, quando il bit A15 é alto, sono indirizzati ingressi-uscite. Ciò corrisponde a posizioni da «8000» a «8FFF» in esadecimale. È da notare che poiché le otto vie di indirizzo a più bassa codifica non sono indirizzate, esse sono «non considerate» nella mappa di I/O presente in memoria.

La prima scheda inizia con l'indirizzo «80XX» (dove «XX» indica che tali bit non hanno importanza) e, poiché ciascun USART ha due registri (ingresso-uscita e controllo), gli indirizzi terminano con «87XX» esadecimale. La piastra successiva copre da «88XX» a «8FXX», e così via, fino all'ultima che va da «B8XX» a «BFXX». Gli indirizzi di pagina pari indicano i registri di stato, mentre gli indirizzi dispari sono per i registri di dati d'ingresso e di uscita.

C'é anche una PROM speciale nella piastra, che é decodificata da un decodificatore speciale. Il suo indirizzo é «70XX» per la prima piastra e «77XX» per l'ultima. La funzione di questa PROM é quella di porre nel bus dei dati l'indirizzo dello USART che ha ricevuto un carattere dal terminale. Come é realizzato? Ciascuna delle vie «RxRdy» nell'USART indicano se é stato ricevuto un carattere. Queste quattro vie, una per ciascun USART, sono connesse alle *vie di indirizzo* della PROM.

Uno dei 16 possibili byte deve essere selezionato mediante decodifica. Il quinto bit di indirizzo é ad un uno o ad uno zero. In tal modo la stessa PROM può essere usata per la piastra zero o per la piastra uno, ponendo nelle altre 16 posizioni gli indirizzi per la piastra 1 e ponendo ad uno il ponticello nel quinto bit di indirizzo. (Ponticello a zero per pari, uno per dispari). Cosa sono queste 16 posizioni? Esse sono semplicemente una tabella degli indirizzi «81», «83», «85» e «87» in esadecimale per la piastra zero e «89», «8B», «8D», «8F» per la piastra uno, PROM simili sono codificate per le altre sei piastre.

I valori sono scelti in tal modo che la prima posizione nella PROM sia un byte di zeri. In tal modo, quando nessun USART ha un carattere, e tutte le vie RxRdy sono a livello basso, il byte di stato é tutti zeri, indicando che non c'é nulla da fare in

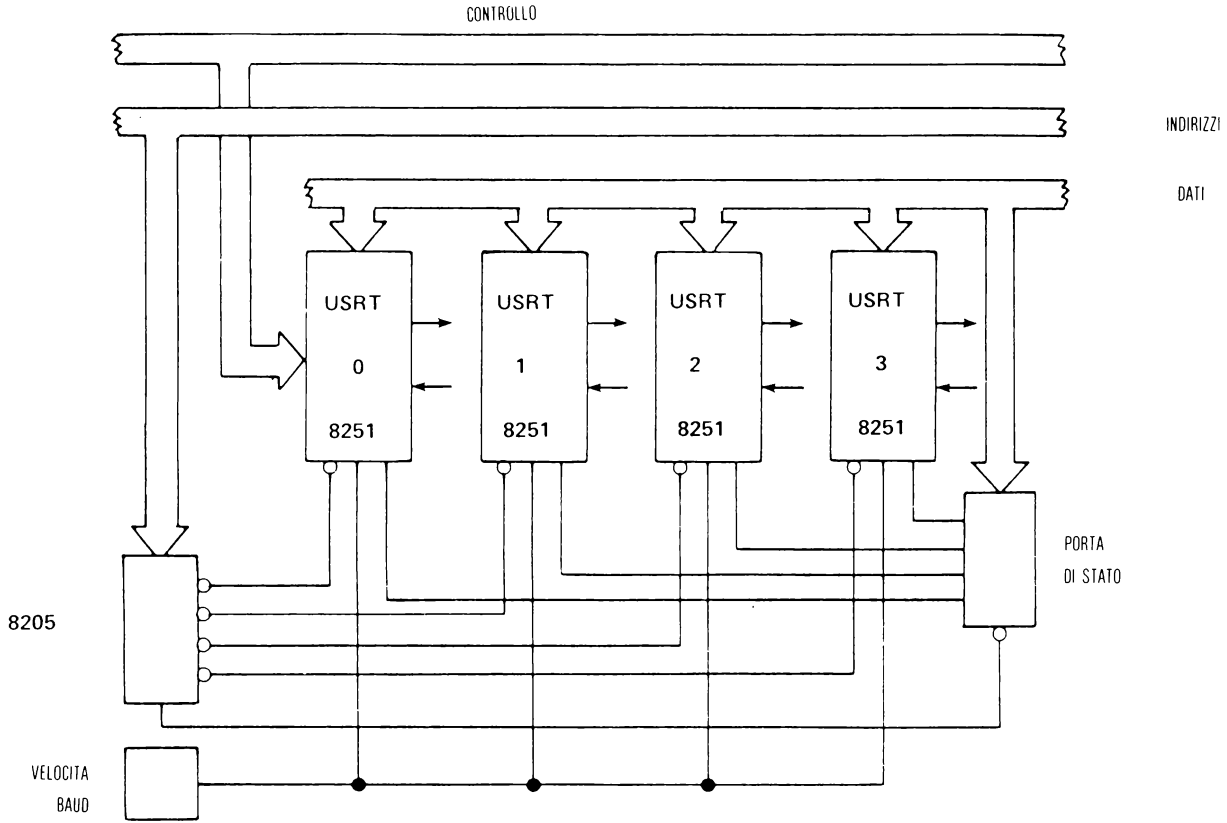


Fig. 7-10: Piastra USART

quella piastra. Se tale byte non é zero, un carattere é in attesa. Per assicurare che é semplice indicare quale USART é in attesa, basta osservare che la posizione successiva contiene il valore «81»: se il primo USART è in attesa, e tutti gli altri non lo sono, il programma riceverà un «81» dalla PROM di stato. Il programma può allora usare tale valore per indirizzare direttamente il carattere attualmente in attesa. Per di più, il valore «81» può essere mascherato, per formare un'etichetta per il dato prelevato.

Le due posizioni successive contengono «83», le quattro successive «85», e le altre otto «87». In tal modo é realizzata una tabella di priorità in modo che, quando è servito ciascun USART, quello successivo in attesa é servito di seguito a sua volta.

Questo metodo di indirizzare la PROM di stato permette al programma di utilizzare soltanto alcune istruzioni per individuare quale USART, tra le 32 possibili, é pronto con un carattere, per prelevare il carattere e per generare la etichetta opportuna in base alla informazione di stato. Ci sono due moduli di interfaccia per gestire gli ingressi e le uscite seriali TTL dell'USART e convertirle in impulsi seriali a +12 e - 12 volt compatibili con la interfaccia EIA RS232C. Essi sono semplici circuiti integrati traslatori di livello.

L'ultima sezione consiste di un multivibratore astabile, sincronizzato da un cristallo per fornire la temporizzazione per i clock seriali dei bit. Su ciascuna piastra ci sono due semplici divisori per fornire agli USART tutte le comuni velocità seriali. Ciò é mostrato in figura 7-11.

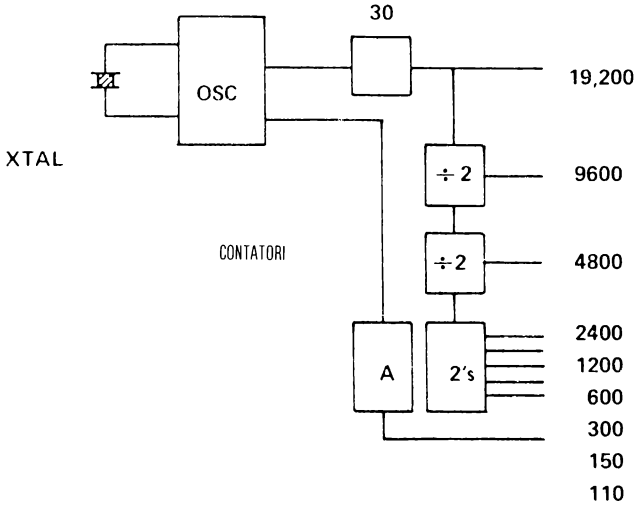


Fig. 7-11: Generatore di velocità in baud

La piastra di interfaccia verso l'elaboratore

Questo modulo contiene: l'USART dell'elaboratore, il controllore di interruzione, e un generatore di velocità in baud per le velocità di comunicazione dall'elaboratore verso il multiplatore. Ciò é indicato in figura 7-12.

Le unità di questa piastra sono indirizzate come porte di ingresso-uscita, invece che posizioni di memoria. L'USART é indirizzata come porta «F9» e «FA» in esadecimale, rispettivamente per i controlli e i dati. Il circuito generatore di velocità é qui duplicato per generare i segnali «TxC» e «RxC» per l'USART dall'elaboratore verso il multiplatore, poiché tali velocità possono differire tra loro in un sistema tipico.

Il controllo di interruzioni preleva i segnali «RxRdy» e «TxRdy» dall'USART, e genera due vettori di interruzione, numero 1 e numero 7. Il numero 1 serve per segnalare che é stato ricevuto un carattere dall'elaboratore e deve essere elaborato, mentre il numero 7 indica che l'USART può essere impegnato nuovamente per trasmettere un'altro carattere all'elaboratore.

Il controllore di interruzioni 8529 é posizionato dalla routine di inizializzazione, per richiamare le routine di servizio nelle posizioni opportune e servire le interruzioni secondo una rotazione ciclica. Dopo che é stata servita una interruzione il software ripristina il bit-semaforo corrispondente nell'8259 e procede con il polling, finché arriva una nuova interruzione.

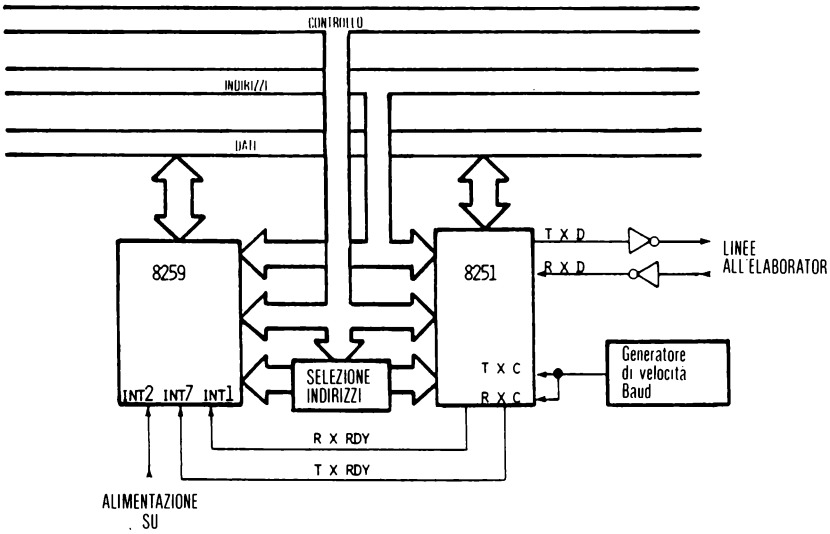


Fig. 7-12: Piastra di interfaccia verso elaboratore

La figura 7-13 mostra la procedura di inizializzazione del PIC e la figura 7-14 mostra il codice di gestione dell'interruzione nelle prime posizioni di memoria.

Il canale verso l'elaboratore é stato fissato a 9600 baud in entrambe le direzioni. I caratteri da ciascun terminale devono essere rinviati in eco. Ciò é possibile poiché il sistema é full - duplex. Per ciascun carattere generato, l'elaboratore deve eseguire la elaborazione e rinviare l'eco. 24 dei terminali sono Lear-Siegler ADM-3s, posti a 9600 baud in ingresso e in uscita. Sono anche presenti quattro terminali a 300 baud e quattro linee a 300 baud dial-up sul multiplatore.

Le porte F7 e F8 sono PIC

CONTROLLO	INDIRIZZO	DATI	OPERAZIONE	
Scrive I/O	F8	32	«settare» gli indirizzi low per la chiamata	per INT 1
Scrive I/O	F7	00	«settare» gli indirizzi high per la chiamata	
Scrive I/O	F8	F2	«settare» gli indirizzi low per la chiamata	per INT 7
Scrive I/O	F7	00	«settare» gli indirizzi high per la chiamata	
Scrive I/O	F7	70	abilita solo INT 1 e 7	
Scrive I/O	F8	A0	«settare» la priorità rotante nel modo reset	

Fig. 7-13: Formato di impegno in software di un PIC

Tipica velocità media di ingresso é dieci caratteri al secondo. La velocità media di uscita é 200 caratteri al secondo. Le memorie tampone nell'elaboratore, per i caratteri in attesa verso il canale di uscita sono vuote per il 95% del tempo, indicando che l'elaboratore può liberarsi dei dati alla stessa velocità con la quale sono gestiti dal canale, piuttosto che tanto velocemente quanto i terminali possono stamparli. Massime velocità misurate cono 15 caratteri al secondo in ingresso, e 620 caratteri al secondo in uscita. I valori massimi e medi sono stati valutati in un periodo di 17 ore, durante il quale il 90% dei terminali erano funzionanti.

I tassi di errore sono stati interamente dovuti al canale, o quanto meno altri errori, di operatore, dell'elaboratore, o di altro, non erano distinguibili.

Nelle figure 7-16, 7-17, 7-18 e 7-19 sono indicate fotografie delle piastre di circuito stampato.

```
0000                ORG OH          Start di inizializzazione
0000  00     RST0:  NOP
0001  31FF2F        LXI SP,2FFFH    Poni il puntatore dello stack
0004  F3            DI              Disabilita le interruzioni
0005  C3D700        JMP INIT        Ripartenza del sistema dopo il reset
0008  C5     RST1:  PUSH B          Vettore RST dall'elaboratore verso il mux
0009  D5            PUSH D          Vettore di stato PUSH
000A  E5            PUSH H
000B  F5            PUSH PSW
000C  CD4900        CALL INT70      INT 70 presenta il carattere dall'elaboratore
000F  3E08          MVI A,0008H     Decodifica e riprendi attività
0011  D3F8          OUT 00F8H       Il controllore di interruzione ripristina
                                    l'interruzione

0013  F1            POP PSW
0014  E1            POP H
0015  D1            POP D           Semaforo
0016  C1            POP B           Vettore di stato POP
0017  EF            RST 5           Coda primaria
0018  FB            EI

0019  C9            RET
0020                ORG 0020H
0020  CDC700 RST4:  CALL SND50      Reset software
0023  C7            RST 0

0028                ORG 0028H
0028  F5     RST5:  PUSH PSW        Salva A e i semafori
0029  DBFA          IN 00FAH        Leggi lo stato dell'USRT
002B  E601          ANI 0001H       CHK per TXRDY
002D  CA3100        JZ POPAF        Se USRT è impegnato riprendi
0030  FF            RST 7           Altrimenti chiama RST 7 per servizio FIFO
                                    Per CHK se qualcosa è nella
                                    FIFO da spedire all'11/70
0031  F1     POPAF: POP PSW
0032  C9            RET
0038                ORG 0038H
0038  C5     RST7:  PUSH B          Vettore RST del MUX verso l'elaboratore
0039  D5            PUSH D
003A  E5            PUSH H          Canale non impegnato
003B  F5            PUSH PSW
003C  CD1802        CALL OINT       OINT pone in uscita il carattere
003F  3E08          MVI A,0008H
0041  D3F8          OUT 00F8H       Dalla coda
0043  F1            POP PSW
0044  E1            POP H
0045  D1            POP D
```

Fig. 7-14: Esempio di controllo di interruzione

RST0;	Inizializza l'hardware.
RST1;	È arrivato un carattere dall'elaboratore.
RST4;	Reset software sul programma di rivelazione guasto software.
RST5;	Verifica di non impegno del canale verso l'elaborazione. La coda del mux verso l'elaboratore è vuota.
RST7;	Verifica di non impegno del canale verso l'elaboratore. Controllo della coda di memoria per i caratteri; se qualcuno è presente trasmetti, altrimenti ritorna al programma.

Fig. 7-15: Vettori realizzati in software

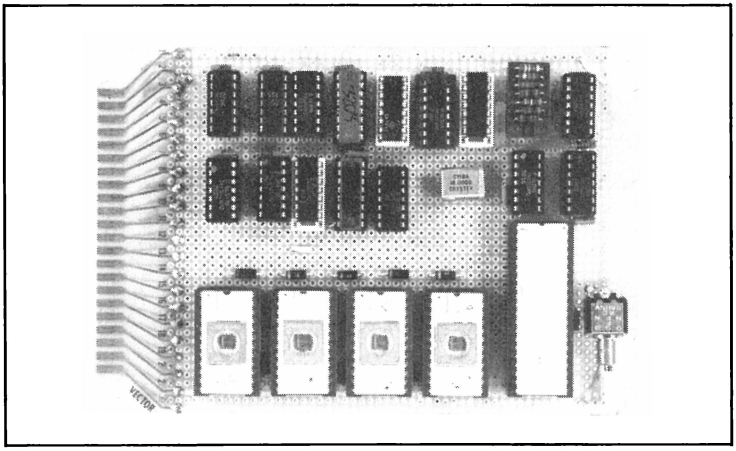


Fig. 7-16: CPU

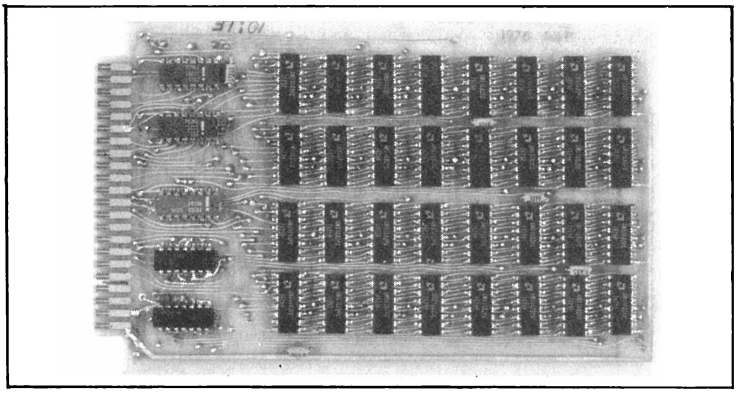


Fig. 7-17: RAM

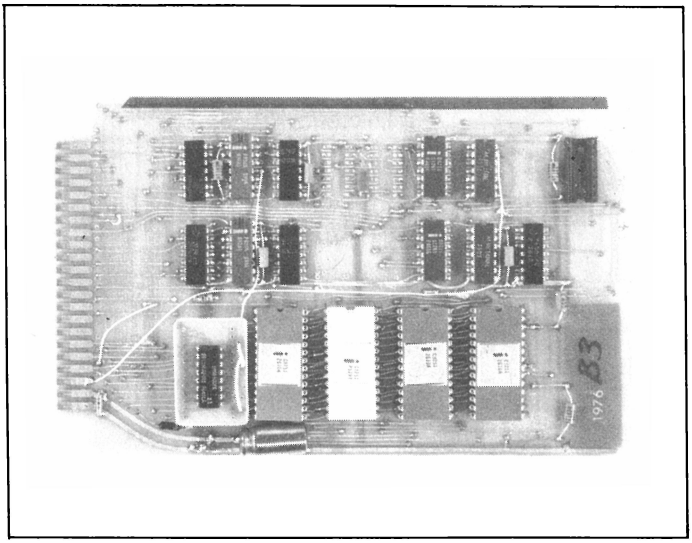


Fig. 7-18: USART di terminale

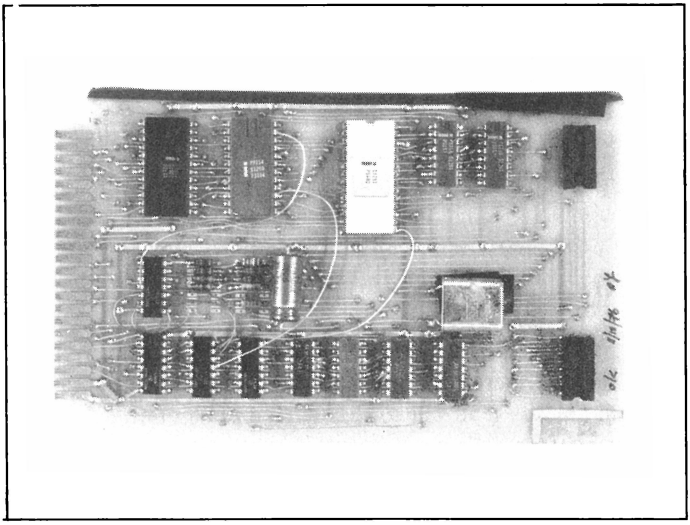


Fig. 7-19: Controllo dell'elaboratore e dell'interruzione

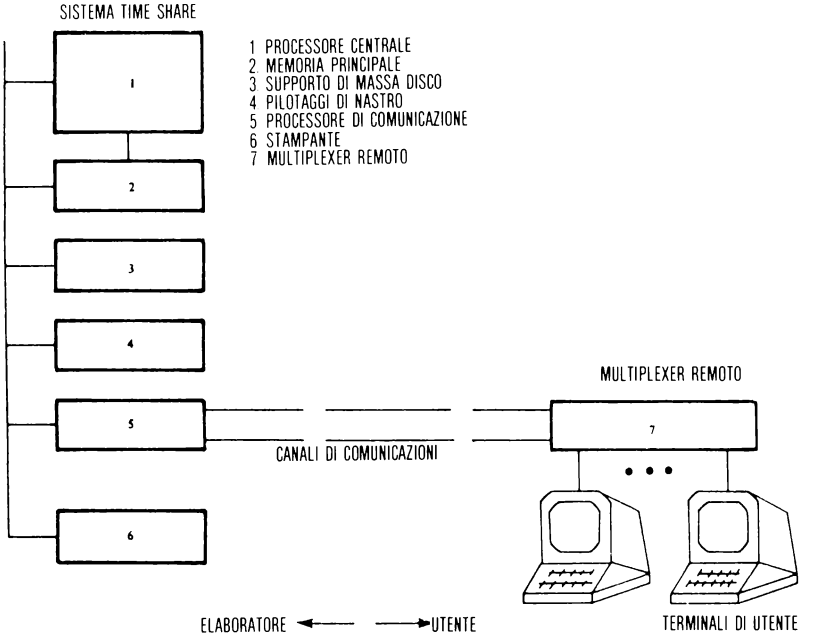


Fig. 7-20: Intero sistema

CONCLUSIONI

In questo capitolo é stata descritta una interfaccia completa. Il modo di procedere passo passo nel descrivere come ciascun componente é stato integrato nel modulo, come i moduli hanno creato un sottosistema e poi l'intero sistema, permetteranno al lettore di procedere nelle applicazioni di molti altri tipi di interfaccia di microprocessore.

Questa particolare applicazione utilizza gran parte delle tecniche discusse nei capitoli precedenti: interruzioni, gestione della memoria e dell'I/O, comprendenti tecniche speciali per la riduzione dell'hardware mediante il software e interfacce verso componenti esterni.

CAPITOLO 8

PROVE

INTRODUZIONE

Cosa si fa quando un'apparecchiatura non funziona? Cosa è accaduto e perchè? L'attività di ricerca degli errori, conosciuta anche come testing o trouble-shooting è una parte integrante del progetto di un sistema. La legge di Murphy afferma: se qualcosa può guastarsi, certamente lo farà. Quando si è di fronte ad un sistema che si comporta non correttamente, vi sono un certo numero di tecniche per identificare e correggere il problema.

In questo capitolo saranno presentate le cause dei problemi più comuni e la loro soluzione. Problemi quali: rottura di componenti, malfunzionamenti di software, malfunzionamenti indotti da rumori, saranno analizzati e saranno presentati dei metodi per identificarli.

Saranno inoltre descritte le apparecchiature necessarie per identificare e localizzare tali problemi, quali: voltmetri, sonde logiche, analizzatori di valori caratteristici, oscilloscopi analizzatori digitali, emulatori di circuiti interni, emulatori e simulatori.

Alla fine sarà presentato il caso di «un bit su 16.384». L'esempio illustra la fase di ricerca degli errori sul progetto del multiplexer presentato al capitolo 7.

COS'È CHE NON FUNZIONA?

In un sistema i problemi principali che possono esserci sono: guasti alla circuiteria — interruzioni o corticircuiti; guasti ai componenti — includendo i componenti mal dimensionati; errori di software; rumori ed interferenze sia interne che esterne.

I *guasti alla circuiteria* si possono rilevare con una semplice verifica di resistenza da punto a punto del sistema.

Ci si deve assicurare che ciascun filo vada a finire sul corretto attacco del circuito integrato. Assicurarsene due volte.

Non essere mai sicuri che lo schema circuitale non presenti errori finchè non funziona.

Gli errori alla circuiteria sono i più comuni ed anche i più fastidiosi. Essi possono essere risolti facilmente, ma portano via del tempo.

La maggior parte dei circuiti sono provati con un «cicalino» che emette un breve tono se c'è continuità e non emette nulla se il circuito è interrotto. Questo tipo di strumento permette di lasciare liberi gli occhi dal controllare il circuito.

Guasto dei componenti

Componenti quali resistenze, capacità, induttori, trasformatori, transistor, diodi, circuiti integrati e connettori possono presentare ogni tipo di guasto.

Le resistenze si interrompono, le capacità perdono l'elettrolitico.

In breve si può dire che *non esiste componente perfetto, tutti prima o poi si rompono.* Di ciascun componente si dà una figura di merito, conosciuta usualmente come la sua *mean-time-betwen-failure* o MTBF. Essa è una previsione statistica, espressa in ore *di quanto il componente durerà in un dato ambiente.*

In figura 8-1 sono riportati per applicazioni di avionica militare i vari tassi di rottura percentuali dei singoli componenti riferiti a 1000 ore di funzionamento.

COMPONENTI	TASSO DI GUASTO (% 1000 h)
1. Capacità	0,02
2. Contatti di connettore	0,005
3. Diodo	0,013
4. Circuiti integrati SSI, MSI, LSI	0,015
5. Cristalli a quarzo	0,05
6. Resistenze	0,002
7. Punti di saldatura	0,0002
8. Trasformatore	0,5
9. Transistore	0,04
10. Resistenza variabile	0,01
11. Punto di saldatura «wire-wrapped»	0,00002

Fig. 8-1

Alcune parti in media durano più di altre. Naturalmente in questa tabella si ammette *che tutte le parti siano usate in modo corretto.* Tali valori sono basati su prove accelerate di vita su numerosi esemplari per ogni componente.

Il tasso di guasto è definito come 1/MTBF. Conoscendo il tasso di guasto di ciascun componente in un sistema si può ottenere il tasso di guasto dell'intero sistema. La regola è di sommare tutti i tassi di guasto di tutti i componenti del sistema. Questo numero fornisce il tasso di guasto dell'intero sistema, l'inverso ci fornisce l'MTBF del sistema.

Per esempio supponiamo di avere tre circuiti LSI integrati, un cristallo, dieci resistenze, dieci condensatori, una piastra a circuiti stampati con connettori, un trasformatore, quattro diodi ed un regolatore di tensione I-C.

Il sistema è usato nelle stesse condizioni in cui i vari componenti sono stati provati.

Quale è il tasso di rottura del sistema?
Usando la figura 8-1 si trova:

quattro IC	0,06	
cristallo	0,05	
dieci resistenze	0,02	
dieci condensatori	0,50	
piastra con circuiti stampati	0,60	(10 connettori, 500 punti di saldatura)
trasformatore	0,50	
diodi	0,052	
Totale	1,82%	/1000 ore

Il sistema ha un MTBF pari a:

$$1 \;/\; 1{,}82\% \;/\; 1000 \text{ ore} \simeq 60.000 \text{ ore}$$

Supponiamo di avere costruito 1000 di questi sistemi e di usarli nello stesso ambiente. Dopo 1000 ore è molto probabile che 18 di questi si siano guastati. Dopo 10.000 ore 180 sistemi saranno guasti.

Quanto spesso si rompe un componente? Si è risposto a questa domanda in termini di media dei guasti ma non abbiamo detto niente in termini di *distribuzione dei guasti stessi.*

La maggior parte dei componenti ha un tempo di vita del tipo mostrato in figura 8-2.

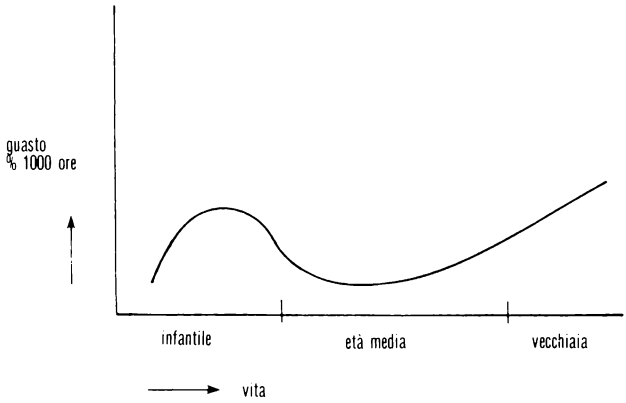


Fig. 8-2: Percentuale di guasto su 1000 ore in relazione all'anzianità

La maggior parte dei guasti avviene quando il componente è nuovo o quando è vecchio e solo pochi guasti avvengono nell'età media.

Il significato di «vecchio» e «nuovo» differisce per ogni componente. Un'analisi profonda implica calcoli lunghi riguardanti la storia dei guasti durante la vita di ciascun componente.

In genere si cerca di eliminare la *«mortalità infantile»* prima che i componenti vengano immessi sul mercato. La tavola precedentemente mostrata è valida solo per uno specifico ambiente.

Applicazioni commerciali, industriali, militari, richiedono modi diversi di misurazione dello MTBF. Un'apparecchiatura usata sui giocattoli può durare 5 anni se usata per lo scopo per cui è nata, se si porta nello spazio essa non durerebbe che 5 minuti.

L'ambiente di applicazione determina quale tipo di durata statistica deve essere usata.

Fino ad ora abbiamo solo toccato l'argomento relativo all'*affidabilità*. Un altro problema è la *qualità*. Contrariamente all'intuizione un'alta qualità non significa sempre un'alta affidabilità. La qualità si riferisce a come un componente si comporta durante il suo lavoro. Il componente può essere rumoroso, dissipare una quantità elevata di calore, ma può durare più a lungo di un altro che è silenzioso e dissipa poco. Solo attraverso una analisi statistica si può determinare l'affidabilità. La qualità può essere facilmente misurata elemento per elemento.

Software

Il software può avere errori. Supponiamo per esempio di avere in programma una speciale routine che gestisce la caduta di alimentazione. Supponiamo anche che sia stato fatto un errore nel programma che ristabilisce la funzionalità della macchina quando ritorna l'alimentazione.

Se questa routine non è mai stata provata essa non sarà mai usata fin quando non cade l'alimentazione. Solo allora ci si accorge che la macchina non risponde alle specifiche.

Un secondo esempio può essere un calcolo aritmetico che causa overflow e quindi condizione di blocco quando l'ingresso misurato è «0». Il sistema può lavorare bene per mesi e poi iniziare a bloccarsi misteriosamente ogni due giorni. I problemi di software o *«bugs»* sono i più complicati da risolversi. In genere la soluzione viene trovata dopo accese discussioni fra tecnici e programmatori. Tuttavia i problemi di software sono i più comuni nei microprocessori. Nessun programma è perfetto. I programmi sono limitati in precisione, velocità e flessibilità. Un programmatore intelligente è sempre pessimista circa il suo programma fin quando non abbia lavorato correttamente per anni.

Disturbi

I disturbi sono presenti ovunque. Ogni volta che in un filo c'è una corrente c'è anche un campo elettromagnetico. Pertanto campi elettromagnetici causati da tra-

sformatori di potenza, motori, fili elettrici in genere sono presenti ovunque.

In più in presenza di radio, televisioni, apparecchiature per radio amatori, ogni filo diventa un'antenna.

Inoltre i rumori non solo provengono dall'esterno, ma possono essere creati anche all'interno del sistema.

Analizziamo quattro casi:

1. Quando si accendono e spengono circuiti integrati essi generano variazioni di corrente dovute alle variazioni di potenza richiesta. Se molti circuiti vengono accesi o spenti allo stesso istante la tensione di alimentazione può cambiare in maniera tale da influenzare gli altri circuiti. Normalmente vi sono condensatori di bypass su ogni circuito integrato per prevenire questo tipo di disturbo.
2. Se due fili sono vicini un impulso che interessa uno dei due induce un impulso anche sull'altro perchè vi è un effetto trasformatore fra i due. L'impulso indotto può modificare un flip-flop o creare un dato incorretto. Per prevenire ciò si usano linee di trasmissione attorcigliate e schermate.
3. L'alimentazione può non essere progettata correttamente. Vi è una certa quantità di ondulazioni a 50 Hz nell'alimentatore a 5 volt. Questo può variare il contenuto della memoria e causare una errata lettura o scrittura. Un'alimentazione correttamente progettata tiene conto delle cadute di tensione che si hanno prima del circuito di regolazione in condizioni di pieno carico.
4. In figura 8-3 vi è un picco di disturbo creato da una teletype quando viene accesa. Si noti, in figura 8-4, cosa accade in una linea di alimentazione senza filtri antidisturbo.
   Se questo accade in momenti particolari si perdono dati e la macchina sbaglia.

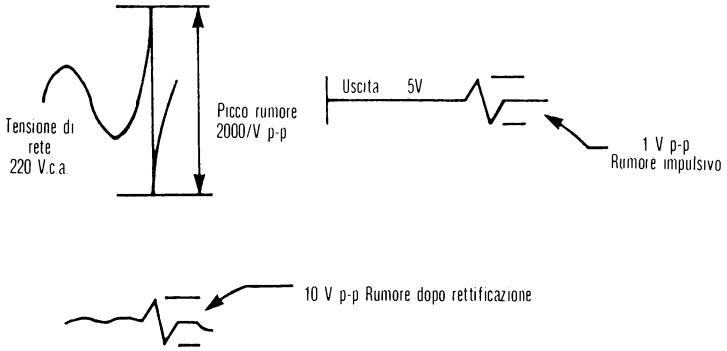


Fig. 8-3: Rumore impulsivo sulla linea di alimentazione

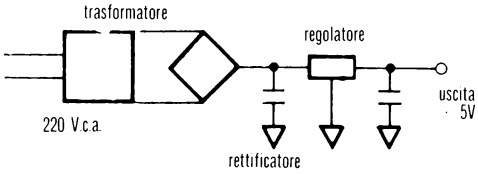


Fig. 8-4: Alimentatore senza filtro di linea

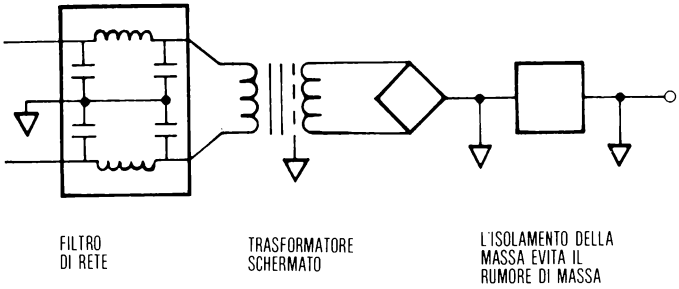


Fig. 8-5: Alimentatore con filtro di linea

La soluzione è nell'usare un filtro di linea e schermare il trasformatore allo scopo di prevenire gli impulsi ad alta frequenza, come si può vedere in figura 8-5.

Sommario dei guasti comuni

I componenti si guastano a tassi prevedibili, il software può non essere affidabile e corretto ed i disturbi possono essere generati all'interno ed all'esterno del sistema.

Come si possono trovare i guasti in maniera razionale? Qui di seguito si indicheranno gli strumenti usati per trovare i guasti ed identificarli. Nella discussione degli strumenti saranno discussi i particolari di ogni problema.

Strumenti per individuazione dei guasti

Qui di seguito saranno presentati gli strumenti ed i tipi di problema che possono risolvere. Tali strumenti saranno esaminati con le loro limitazioni.

La figura 8-6 presenta un breve sommario dei problemi e degli strumenti atti a risolverli.

Ciascun problema sarà considerato secondo l'ordine della figura e sarà indicato come lo strumento può individuarlo e quanto tempo può richiedere.

MATRICE DI DEBUG: PROBLEMI E STRUMENTI

Si dispone delle apparecchiature

Si possono risolvere problemi come:	VOM	PROBES	SGN.ANA.	OSC.	D.D.A.	I.C.E.	EMU.
• Corti, interruzioni tensioni errate	si	possibile	no	si	possibile	possibile	no
• Resistenze, capacità errate	si	no	no	si	no	no	no
• Segnali logici non noti, catena di guasti prodottisi	si	si	si	si	si	no	no
• Segnali logici non noti, catena di guasti potenziali	si	si	no	si	si	si	no
		richiedono molto tempo		richiedono molto tempo			
• Problemi software	no	no	no	possibile	si	si	si
Per individuare un problema tipico		Sono almeno necessari					
Valutazione	si	si		si			
In un tempo medio	si			si	si		
Modo più veloce possibile	si			si	si	si	

TABELLA DELLE ABBREVIAZIONI

VOM	VOLT-OHM-MILLIAMPEROMETRO
PROBES	SONDE LOGICHE
SGN.ANA.	ANALIZZATORE DI SEGNALI
OSC.	OSCILLOSCOPIO
D.D.A.	ANALIZZATORE DI DOMINIO DIGITALE
I.C.E.	EMULAZIONE IN FUNZIONAMENTO
EMU.	EMULAZIONE SOFTWARE O SIMULAZIONE

Fig. 8-6: Problemi e strumenti

Problemi semplici

Conduttori in cortocircuito od interrotti, tensioni errate sono i problemi più comuni. Fortunatamente sono anche i problemi più semplici a trovarsi.

Un qualunque misuratore di resistenza può individuare problemi di cortocircuito od interruzione mentre un voltmetro digitale (DVM) od un volt-ohm-milliamperometro (VOM) sarà sufficiente per verificare correnti e tensioni.

Se si conoscono i componenti impiegati e la loro logica è facile (a parte il tempo impiegato) assicurarsi che tutto vada come deve e che tensioni e correnti siano quelle corrette.

Il VOM

Per misurare il voltaggio, il misuratore è posto in parallelo con l'elemento circuitale. La figura 8-7 mostra la misura della tensione all'uscita del regolatore. Il VOM può misurare facilmente tutti questi tipi di tensione, ma bisogna fare attenzione perchè non misura i rapidi picchi o i rumori.

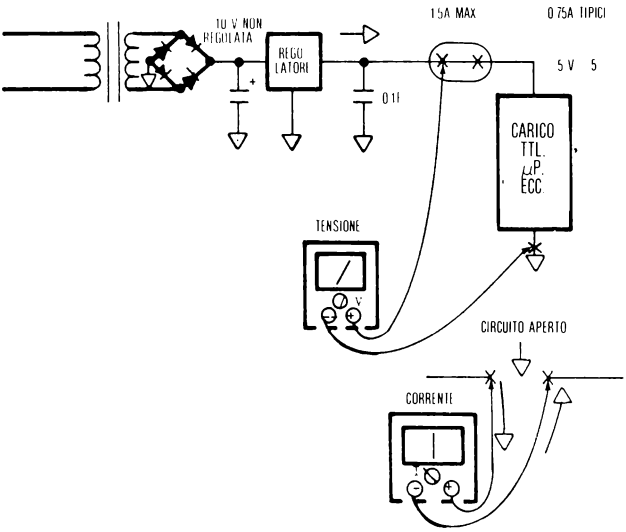


Fig. 8-7: Misura di tensione e corrente con il VOM

Per misurare una corrente il misuratore deve essere posto in serie con il componente. Questo significa che il circuito deve essere interrotto. Possibilmente la connessione deve essere fatta in modo che nelle misure di corrente non si debbano aprire tracce o circuiti.

I comportamenti dinamici di un circuito possono non essere misurabili senza causare dei problemi.

Nell'esempio dell'alimentatore di potenza, il misuratore misurava prima la tensione ai due capi del carico e quindi disconnettendo il carico stesso, riconnettendolo attraverso il misuratore si può misurare la corrente. Bisogna assicurarsi che queste misure siano all'interno della precisione richiesta.

Valori impropri possono causare, in seguito, problemi.

Componenti difettosi

Resistenze, capacità, diodi e transistori possono essere provati confrontandoli con apparecchiature di sicuro funzionamento.

Essi possono essere misurati con i DVM od i VOM per vedere se eseguono almeno le funzioni di base.

Per stabilire le caratteristiche di un diodo o di un transistor occorrono altre apparecchiature speciali.

I circuiti integrati sono difficili da provare senza l'aiuto di apparecchiature costose. Quando si è in fase di prove si deve tenere di scorta una certa quantità di questi componenti in modo da rimpiazzare il componente che ha un funzionamento anomalo.

Una volta che un circuito intero funziona tutti i componenti di scorta debbono essere provati nel sistema prototipo per essere sicuri che in produzione non capitino problemi dovuti alla tolleranza dei singoli componenti. Normalmente piccoli problemi impediscono ad un sistema di funzionare pienamente.

Mal funzionamenti intermittenti sono spesso dovuti ai connettori od a cattive saldature. Questi devono essere provati per primi prima di assumere che il malfunzionamento dipenda da qualcosa d'altro. Tutti i malfunzionamenti intermittenti richiedono l'aiuto di uno oscilloscopio (preferibilmente a memoria) od un analizzatore logico per permettere una veloce soluzione del problema.

Tutti i problemi statici possono essere risolti. Questo è il primo passo: bisogna essere completamente convinti di ciò prima di andare avanti.

Problemi di progettazione

Spesso si sa che cosa si vuole ottenere ma non si riesce ad ottenerlo. Tutti commettiamo errori e lo dobbiamo riconoscere. Gli errori di progetto si dividono in due categorie: *specifiche errate* o *cattivo uso delle stesse.*

Cattivo uso delle specifiche

Una corrente troppo elevata su un registratore causa la sua rottura. Una tensione troppo alta su un condensatore causa il suo cortocircuito. Ogni componente ha i suoi limiti. Il problema dell'«eccessivo» è il più comune. Ad esempio troppi carichi su una singola linea, di uscita possono causare al sistema errori di lettura o scrittura dei dati in maniera intermittente dovuta alle variazioni di temperatura.

Specifiche errate

Se si pensa che una certa parte sia in grado di pilotare 30 carichi quando se ne

possono pilotare solo 20, questa è una specifica errata. Ciò non sarebbe dovuto comparire nelle specifiche. Più subdolamente, la temporizzazione di una certa parte può dare luogo ad incomprensione. Per esempio se gli indirizzi consentiti ad una certa zona di memoria devono essere stabili per 20 nanosecondi prima degli impulsi dei dati di scrittura, questo può essere stato analizzato ma il progettista della temporizzazione può aver violato queste condizioni. I problemi di progetto richiedono una completa gamma di apparecchiature per l'individuazione dei guasti, ma una combinazione VOM-Oscilloscopio sarà spesso sufficiente se il tempo di ricerca del guasto non è molto importante. Questi problemi si manifestano essenzialmente in maniera intermittente in caso di sovraccarico delle bus-line, ed in caso di sovraccarichi di corrente o di tensione.

I problemi di sovratemperatura dei componenti si risolvono aumentando la potenzialità dei componenti stessi o cambiando la progettazione per funzionare con gli stessi componenti.

I problemi di natura intermittente richiedono di provare tutti i carichi dovuti all'input - output, le specifiche dei singoli componenti ed il funzionamento del sistema a diverse temperature, allo scopo di localizzare i componenti più sensibili.

Con l'aiuto di spray refrigeranti o lampade termiche si possono localizzare i problemi dovuti a temperatura mediante riscaldamento e raffreddamento delle parti sospette.

Sonde logiche

Con le *sonde logiche* si possono verificare facilmente i livelli logici in modo da isolare efficacemente tutte le condizioni statiche.

La sonda indicherà se il segnale è a zero, uno, o indeterminato con l'ausilio di un indicatore a LED o di lampade. Nel caso di stati indeterminati si deve fare attenzione: se si ha un bus a tre stati fluttuante, e durante l'esame esso deve essere in tale stato, in realtà esso può essere guasto.

La figura 8-8 mostra l'uso di una sonda logica.

PROBLEMI DINAMICI

Il sistema non funziona. *Il VOM, le sonde logiche etc. non sono correlate con il tempo. Quindi esse sono di scarso aiuto nei casi dinamici.* Occorrono strumenti che indichino che la temporizzazione del livello logico sia corretta.

Oscilloscopio

Per ottenere informazioni sulla temporizzazione l'*oscilloscopio* è il mezzo più comunemente usato. Con l'ausilio di una o più tracce gli eventi possono essere accuratamente misurati in grandezze, durata ed in funzione del tempo. Nei sistemi a microprocessori possono essere osservati con un oscilloscopio eventi fino a 10 nanosecondi. Un'onda quadra di 10 nanosecondi apparirà sull'oscilloscopio a 10 MHz come un'onda sinusoidale. Tuttavia per osservare eventi di questo tipo è desiderabi-

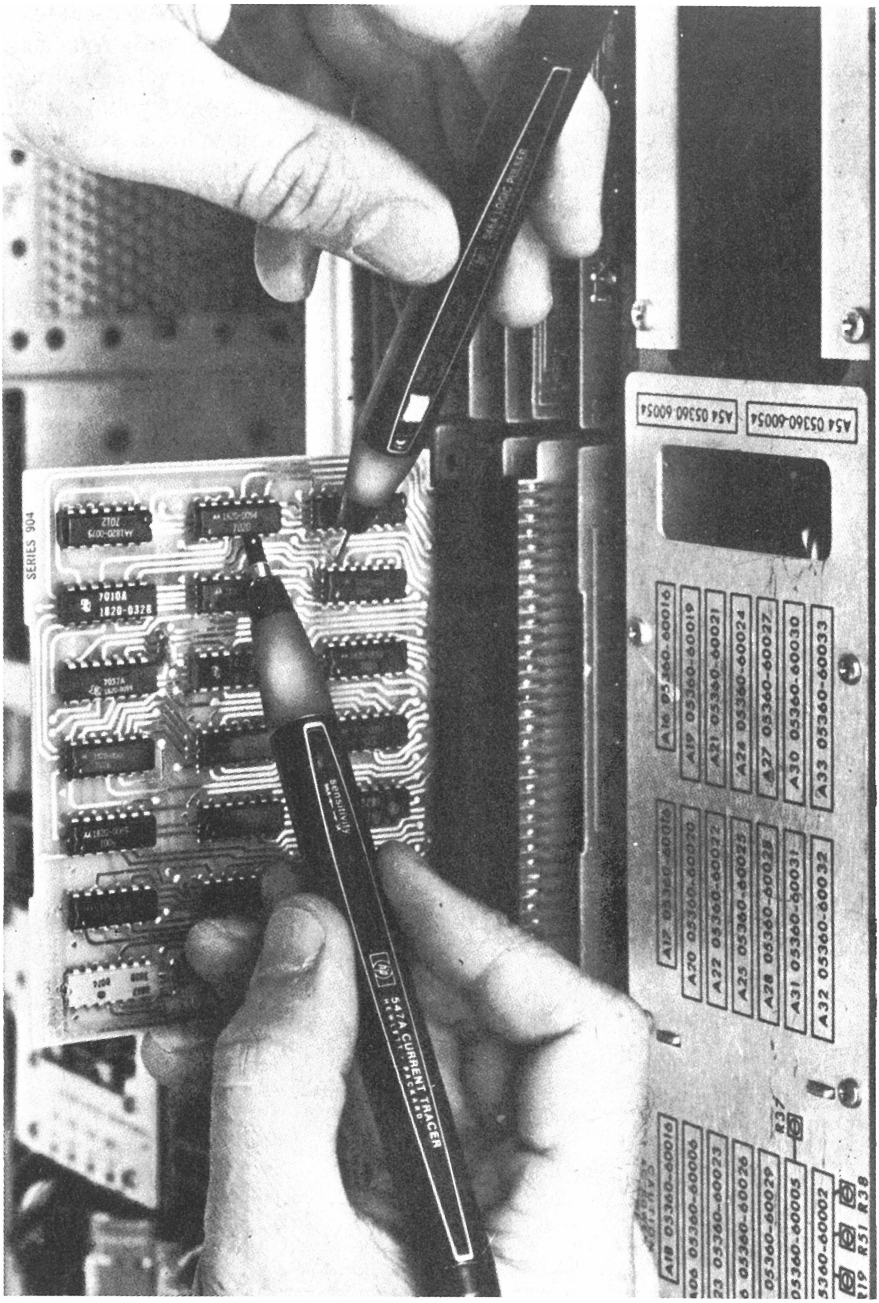


Fig. 8-8: Sonde logiche

le avere uno oscilloscopio a 50 o 100 MHz. La figura 8-9 illustra in un oscilloscopio la traccia di un segnale di controllo logico TTL. La definizione della zona logica per i TTL standard è la seguente: il segnale 0 logico comprende voltaggio fra – 0,6 e +0,8 volt. Il segnale logico 1 è compreso fra + 2,0 volt e + 5,5 volt. La zona fra + 0,8 e + 2,0 volt è considerata indeterminata. La transizione fra un livello logico e l'altro deve avvenire in meno di un microsecondo per evitare problemi di rumore. L'oscilloscopio indicherà se é presente un cattivo segnale logico. Per esempio se 2 uscite TTL sono connesse fra loro, sono state violate le regole di progetto. Se tale condizione permane, una delle uscite può rovinarsi.

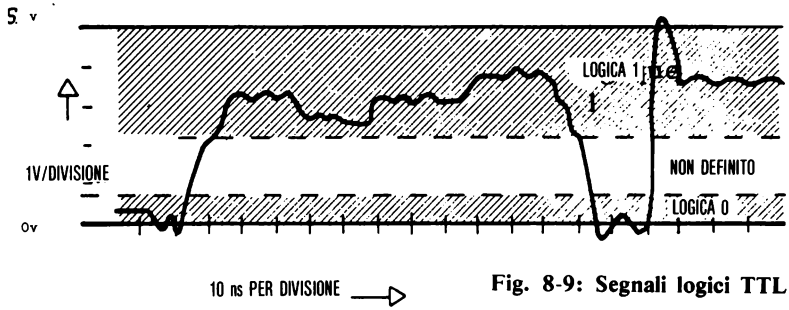


Fig. 8-9: Segnali logici TTL

Se la condizione capita per soli pochi microsecondi alla volta non vi saranno danni; tuttavia l'errore può causare problemi. La figura 8-10 mostra la traccia che

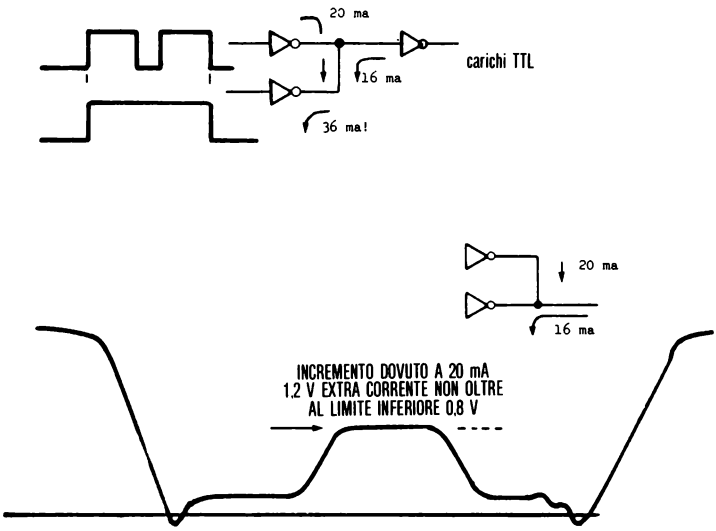


Fig. 8-10: Guasto su uscita TTL

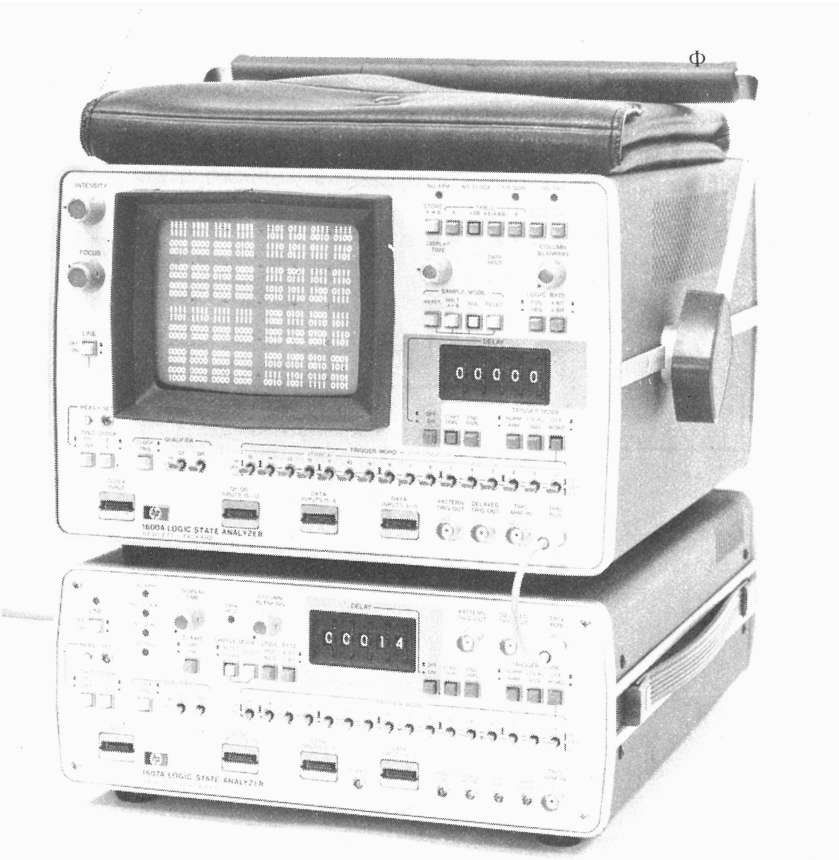


Fig. 8-11: Analizzatore 1600S della HP

se ne ricava in queste condizioni.

Si noti come il livello logico 0 non sia corretto.

Queste misure unite con la conoscenza delle specifiche logiche indicheranno dove è l'errore. Osservando con l'oscilloscopio i chip-select, chip-control, bus-line si individueranno i problemi di carico, di temporizzazione e di rumore. Ci si deve sempre assicurare che i livelli logici siano ben definiti.

Lo 0 TTL deve essere compreso tra -0,6 e + 0,8 volts. Lo 1 TTL deve essere compreso fra + 2,0 e + 5,5 volts. Ogni altra cosa significa che ci sono problemi.

MISURA DI STATO

Tutte le temporizzazioni ed i livelli logici sono corretti quando si osserva un singolo bit od una singola linea; ma spesso si devono osservare tutte le linee allo stesso istante. Si possono riunire insieme 16 oscilloscopi ed in effetti gli analizzatori più vecchi erano dei semplici oscilloscopi a multicanale. Però non è facile osservare 32 minuscole tracce davanti al tubo di un oscilloscopio. Per questa ragione sono stati sviluppati gli *analizzatori logici* o più correttamente *analizzatori a campo digitale.*

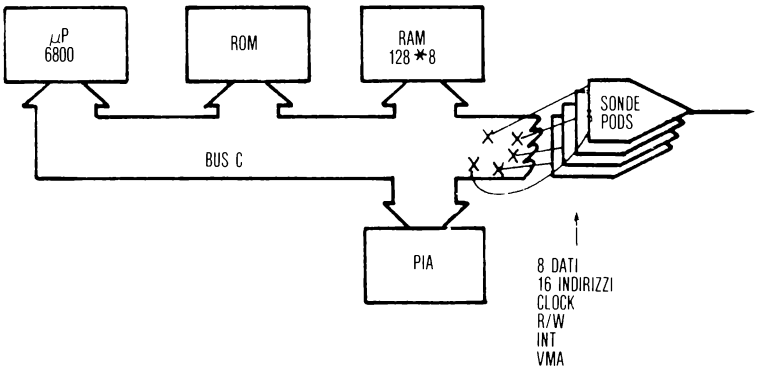


Fig. 8-12: Connessione dell'analizzatore

Analizzatori logici

Cosa permette di fare un analizzatore a campo digitale? Esso permette di osservare fino a 32 bit *simultaneamente.* Tale apparecchiatura visualizza questi bit in binario, ottale, esadecimale e in forma di traccia. Esso comincerà a visualizzare l'informazione quando arriva una data combinazione di bit o «trigger».

Tale apparecchiatura memorizza ad ogni ciclo o più frequentemente una nuova configurazione di segnali ed è anche capace di visualizzare un certo numero di segnali *prima e dopo* l'innesco del trigger.

Gli analizzatori disponibili si dividono in due categorie: quelli che evidenziano le

informazioni di *temporizzazione* e quelli che evidenziano informazioni di *stato*. Gli analizzatori orientati alla rilevazione della temporizzazione sono essenzialmente oscilloscopi multicanale. Queste apparecchiature sono di aiuto dove si sospettano problemi di rumore o livelli logici. Gli analizzatori orientati alla rilevazione degli stati servono ad indicare il flow degli stati del sistema. Tali analizzatori servono ad individuare errori di software e malfunzionamenti complessi di software e hardware.

Esempio di un analizzatore di stato

L'analizzatore Hewlett-Packard 1600S ha 32 canali, 2 temporizzatori, 4 possibilità di innesco e molte altre possibilità. Lo strumento esegue una istantanea dello stato del sistema ogni ciclo del temporizzatore. Si userà l' HP 1600S per osservare il ciclo di interruzione in un sistema basato su un 6800.

La figura 8-13 mostra il formato dei dati visualizzati nel 1600S. Le sonde sono attaccate alle linee indicate. Il temporizzatore è stato connesso a 2.

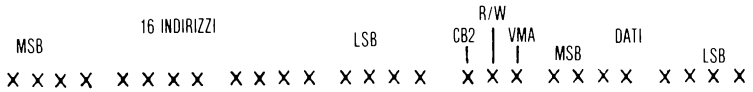


Fig. 8-13: Formato dello schermo dell'HP 1600S per un'interruzione del 6800

Il 1600S è stato innescato su un segnale di interruzione.
In figura 8-12 è visualizzto il succedersi degli stati.

I dati visualizzati sono:

1. È finito il ciclo della istruzione corrente.
   L'istruzione è un «F2» esadecimale alla locazione «1385» esadecimale.

2. Lo stato corrente è salvato nello stack, prima di andare alla routine di servizio delle interruzioni.
   Si nota che lo stack è alla locazione «3FF» esadecimale verso il basso. Il contatore di programma, gli accumulatori, i registri indice ed i flags sono salvati nelle successive locazioni dello stack.

3. Il microprocessore ora carica i contenuti degli indirizzi «FFF 8» ed «FFF 9» esadecimali. I contenuti sono trasferiti nel contatore di programma.

4. La routine di servizio delle interruzioni comincia da «1351» esadecimale. L'esecuzione continua da questo punto.

Con questa apparecchiatura si ottiene una visione dove il sistema era, dove è attualmente e dove sta andando. Alcuni analizzatori memorizzano una particolare

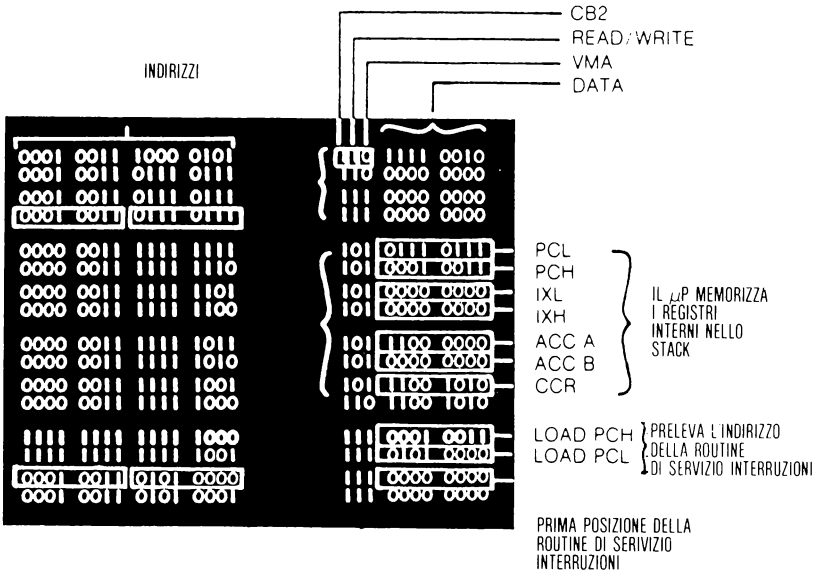


Fig. 8-14: Sequenza di interruzione

sequenza di stati, la comparano continuamente con lo stato corrente e si fermano quando non vi è coerenza.

Altri visualizzano un «1» od uno «0» per ciascun bit in una pagina di memoria ed indicano se quel bit è stato letto o scritto. Alcuni analizzatori memorizzano più informazioni di altri, tuttavia tutti questi analizzatori hanno caratteristiche simili nell'essere capaci di analizzare il numero degli stati di un sistema secondo una sequenza temporale.

L'analizzatore a campi digitali permette al progettista di controllare l'esecuzione software in maniera che possano essere individuati dati errati o istruzioni errate. Se un analizzatore a campo digitale è usato per innescare un oscilloscopio, allora possono essere identificati problemi di rumore e di strane temporizzazioni.

Emulazione del circuito interno

L'emulazione di circuito permette di entrare dentro lo stesso microprocessore e dinamicamente osservare dove esso sta andando e cosa sta leggendo o scrivendo. Ciò rende possibile controllare il processore stesso. Esso include routine di prova e di interruzione in modo da permettere di controllare una specifica sezione del codice e visualizzare il contenuto dei registri interni. Confrontando questo con quello che ci si aspetta, si può localizzre l'errore. Alle figure 8-15 ed 8-16 sono mostrati gli analizzatori della BIOMATION e della HEWLETT - PACKARD.

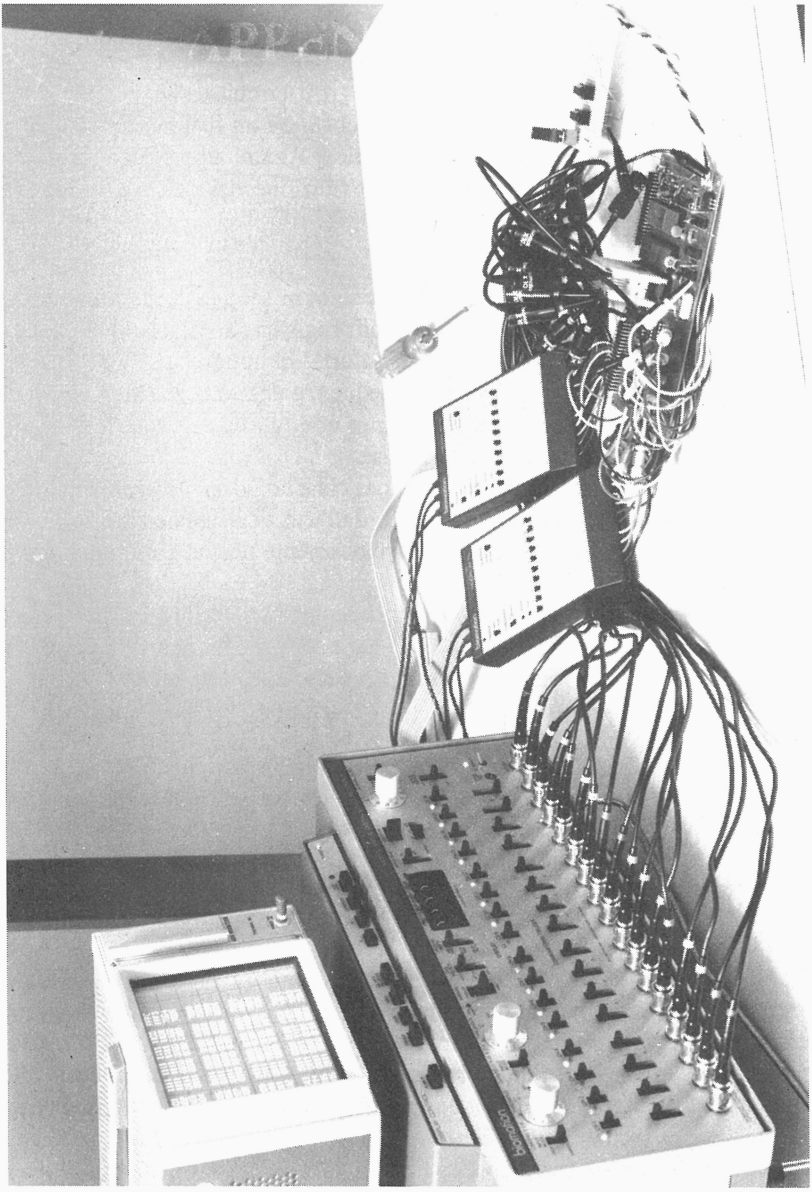


Fig. 8-15: Analizzatore logico della Biomation

Analisi dei valori caratteristici

Vi è una intera gamma di apparecchi speciali usabili solo quando un sistema è stato costruito e provato. Questi sistemi si basano sul funzionamento noto del sistema originale per rilevare gli errori nei sistemi installati. Questa tecnica si basa su un albero di derivazione degli errori. Tutto ciò che potrebbe fare andare male è stato fatto ed in ognuno di questi casi sono stati misurati i nodi del circuito per scoprire come un malfunzionamento si ripercuota lungo il circuito stesso.

Alcuni alberi genealogici degli errori sono corti. Se un fusibile salta, si rimpiazza; se il fusibile salta ancora, si chiama il tecnico. Alcuni alberi degli errori conducono il tecnico attraverso tutto il sistema in funzione dei valori che misura.

Un analizzatore dei valori caratteristici

Questa apparecchiatura si basa sul fatto che ogni sequenza ripetitiva dei valori dei segnali può essere memorizzata in uno shift-register che ricicla, il cui contenuto visualizzato e temporizzato assumerà certi valori. Un apparecchio può essere progettato in modo che la probabilità che due sequenze abbiano gli stessi valori caratteristici è estremamente bassa. Così in ciascun nodo di un sistema avrà la propria «caratteristica» quando lavora correttamente.

Avrà anche una speciale «caratteristica» per ciascun possibile problema. Usando un metodo ad albero degli errori sviluppato dall'uso degli analizzatori, una apparecchiatura può essere riparata arrivando velocemente fino al componente difettoso.

Con questo approccio non si troveranno i problemi software iniziali o la causa dei guasti intermittenti.

In figura 8-18 si vede il diagramma seguito per individuare gli errori utilizzando un analizzatore HP 5004 A.

Queste caratteristiche sono state generate in uno strumento funzionante ed il diagramma è stato sviluppato per incrementare la velocità di riparazione.

TECNICHE SOFTWARE DI PROVA

Il principio comune a tutte le tecniche di prova è di comparare una piastra, componente o sistema con «cosa dovrebbe essere».

Il problema è conoscere cosa dovrebbe essere o anche impostare una procedura per eseguire questo confronto, in maniera sistematica. In altre vi sono due problemi supplementari: fare le misure e memorizzare la storia degli ultimi n segnali.

Per questo scopo sono forniti un certo numero di strumenti hardware e software. Le tecniche e gli strumenti usati per fare questi paragoni sono stati descritti nella sezione precedente.

Come è normale nel mondo dei calcolatori si possono usare sia tecniche hardware che software.

Lo scopo di questo capitolo è di spiegare le tecniche di prova software.

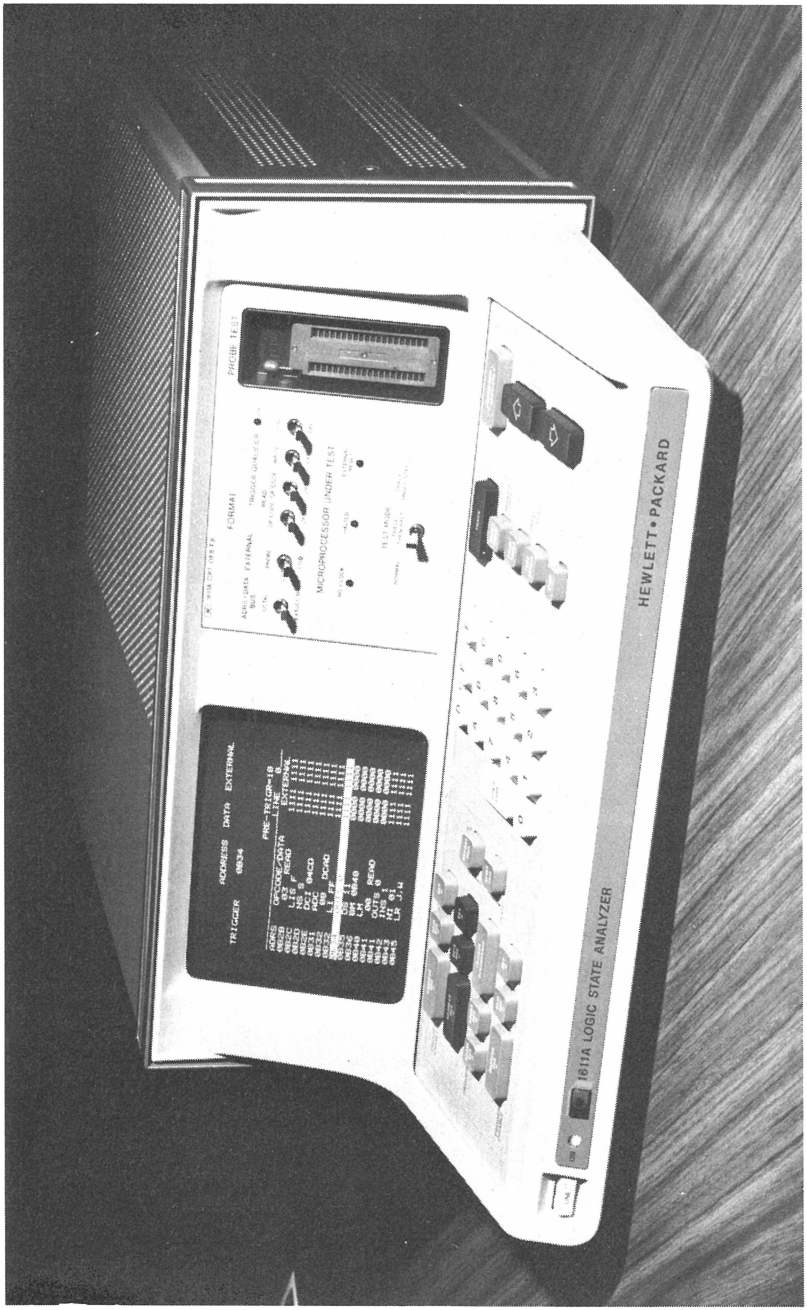


Fig. 8-16: Analizzatore HP ICE per indicazioni mnemoniche del sistema 8080

Fig. 8-17: Analizzatore di segnali HP5004A

Prove di comparazione

In questo metodo una apparecchiatura, o una piastra è comparata rispettivamente con una apparecchiatura «funzionante» o con una piastra «funzionante». Esse condividono l'ingresso e le uscite sono confrontate tra loro.

Questo è un metodo hardware e richiede che le apparecchiature esistano effettivamente.

Le prossime 3 tecniche sono invece essenzialmente software.

Autodiagnostica

Nel metodo di autodiagnostica il microprocessore stesso determina ciò che è funzionante e se non lo è, quale parte del sistema è difettosa. Il principio base dell'autodiagnostica è di eseguire la sequenza del caso «peggiore» ed osservarne i risultati. Nel caso della stessa MPU la sequenza delle istruzioni peggiori è normalmente indicata dal costruttore. Normalmente tale sequenza eseguirà tutte le istruzioni macchina in un ordine prestabilito. In aggiunta può includere alcune sequenze che sono state riconosciute come critiche per la macchina. Chiaramente queste informazioni devono provenire dal costruttore. Molti costruttori sono cooperativi nel fornire tali programmi.

Naturalmente ci si può chiedere: *cosa succede se la stessa MPU è difettosa?* Se essa è difettosa è come se un programma non terminasse correttamente e il sistema si bloccasse senza errori esterni. Quando si esegue questa autodiagnostica si deve usare un sistema automatico di modifiche esterne.

Per esempio il sistema stamperà un messaggio dicendo: «La diagnostica è partita all'istante x».

Al tempo x più un minuto il sistema dovrebbe aver completato l'autodiagnostica e dovrebbe stampare: «Prova di autodiagnostica terminata correttamente». Se questo messaggio non viene stampato si assume che il sistema ha un guasto.

Si possono usare anche apparecchiature esterne. Ad esempio un allarme esterno con propri meccanismi di temporizzazione può essere fatto partire all'inizio della prova. Se entro un certo periodo di tempo il temporizzatore non è resettato, l'allarme segnalerà un guasto nell'MPU automaticamente. Tali programmi di autodiagnostica sono molto usati in sistemi che normalmente sono in uno stato inattivo.

È facile scrivere semplici programmi e farli risiedere in parti della ROM non usata. Quando il microprocessore è inattivo il programma può essere eseguito e verificare l'integrità della macchina. In più se esso viene eseguito continuamente per un certo periodo di tempo aiuterà ad isolare guasti intermittenti nella macchina.

Naturalmente può non risiedere solo in ROM, ma può essere caricato in RAM da una periferica esterna.

L'autodiagnostica è anche usata per provare la memoria e l'input-output. Le prove della memoria saranno viste in dettaglio nel prossimo paragrafo vedendo la generazione algoritmica dei pattern. Nel caso della memoria ROM la più semplice forma di autodiagnostica è detta di *convalida mediante verifica della somma.*

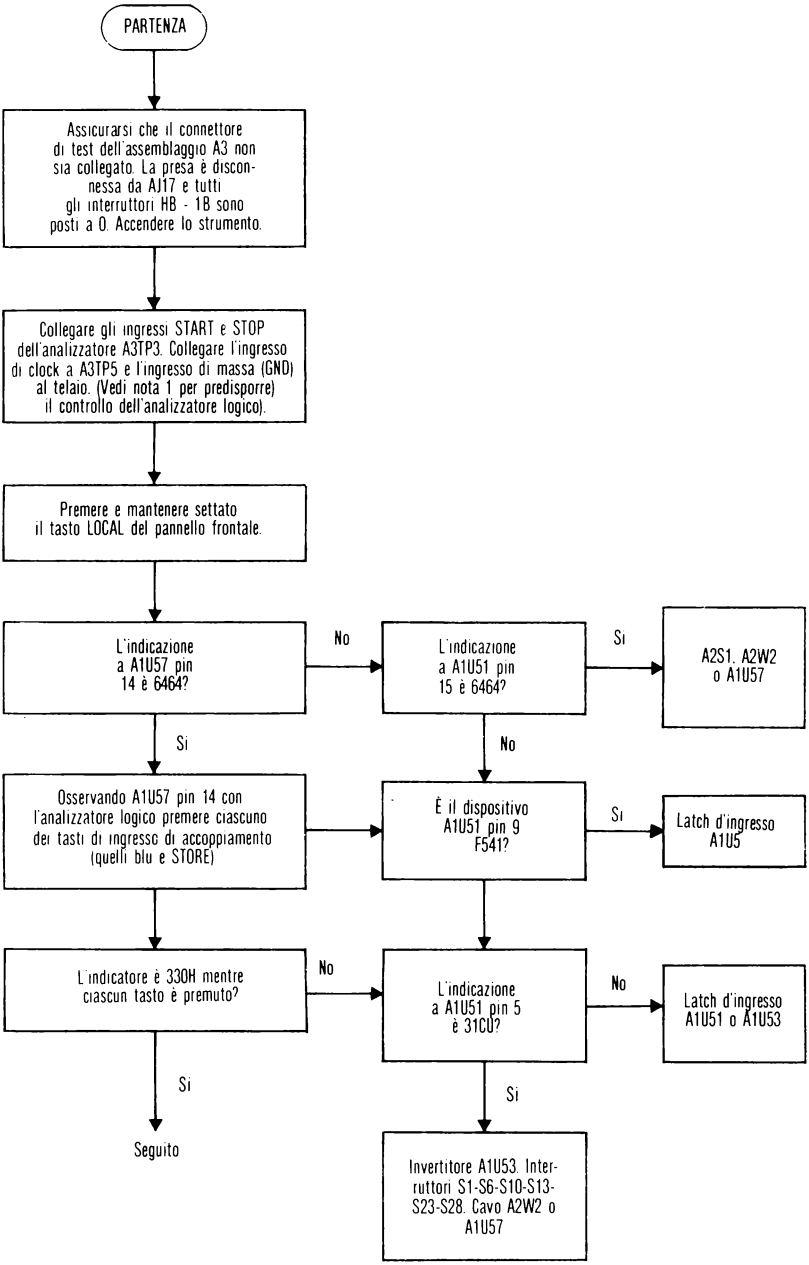


Fig. 8-18: Albero rilevazione guasti

In questa tecnica ciascun blocco di dati di 16,32 o 256 parole è seguito da un byte o 2 bytes per verifica della somma.

Tipicamente questa verifica è eseguita sommando gli n semi-bytes del blocco di n parole usando l'aritmetica esadecimale. Questa somma è quindi troncata agli ultimi 4 digits binari e il carattere di verifica è la codifica ASCII del digit esadecimale risultante.

Un programma eseguito in una area di ROM (che si assume essere funzionante) può leggere il contenuto del resto della ROM, ricalcolare il carattere per la verifica della somma e lo compara con il valore che vi è stato registrato. Se non coincide è stato identificato un guasto della ROM.

La prova delle interfacce di input-output e delle periferiche di I/O normalmente è complessa a causa delle relazioni di temporizzazioni che vi sono coinvolte. Tuttavia è possibile una verifica del corretto funzionamento della periferica stessa.

Assunto che le informazioni di ritorno dalla periferica sono disponibili, il programma eseguirà un ordine del tipo «chiudi relay A». Ammesso che la risposta di ritorno sia disponibile, si può verificare la chiusura del relay entro n millisecondi.

Il sistema può così rallentare tutte le periferiche esterne e provare la loro funzionalità. In più, durante il funzionamento del sistema «prove di coerenza» possono essere eseguite in tutte le periferiche di output (vedi il libro C 201 per una completa descrizione). Queste prove comparano i valori dei parametri di ingresso con i valori di una tabella in memoria e determinano se questi dati sono «coerenti». Per esempio quando si misura la temperatura dell'acqua la temperatura sotto 0° C e sopra 100° C è «incoerente».

Similmente per un microprocessore che controlla i semafori ad un crocevia rilevare la velocità dei veicoli sopra i 300 km/h risulterebbe irragionevole.

Naturalmente in uno specifico ambiente le prove possono essere più raffinate degli esempi precedenti. Queste prove di coerenza rileveranno dei guasti permanenti ed intermittenti delle periferiche di ingresso e faranno scattare un allarme esterno.

Memorizzazione della risposta

Nel metodo di memorizzazione della risposta un computer di elevate dimensioni viene usato per emulare o simulare la periferica o la piastra sotto esame.

Per primo si usa un programma per misurare le caratteristiche della periferica od il sistema in prova preferibilmente sotto un funzionamento dinamico. Questi dati sono registrati e saranno poi usati dal programma di comparazione.

Il programma di comparazione è quindi applicato alla apparecchiatura in prova. Esso genera i segnali di ingresso. L'uscita sarà misurata e comparata con le precedenti risposte del sistema memorizzate nelle tabelle.

Per questo metodo sonó necessarie 2 fasi.

La prima fase è una fase di caratterizzazione dove il computer è usato per memorizzare le risposte essenziali che poi saranno usate in seguito come termini di riferimento.

Una volta che tali risposte sono state ottenute nella fase 2, il sistema funzionerà

solo per confronto, eseguendo un programma di prove specifico e misurandone le risposte.

Questo metodo è usato essenzialmente nella produzione e per prove di accettazione. Il costo di un sistema che esplichi le funzionalità suddette, compreso il programma, può andare da 50.000 a 500.000 dollari.

Generazione algoritmica di pattern

Una generazione algoritmica di pattern è usata essenzialmente per verificare la memoria RAM. Il principio consiste nello scrivere un pattern in memoria e quindi verificare che:

1. sia stato scritto correttamente e che
2. nulla sia stato scritto in nessuna altra parte della memoria a causa di malfunzionamento della RAM.

Le due tecniche di generazione dei pattern usate per provare le RAM sono: tecniche a pattern fissi e tecniche di pattern.

Prove con pattern fissi

In una prova a pattern fissi si scrivono e si rileggono in ogni locazione di memoria pattern identici, alternati o ciclici. Questa tecnica permette di individuare dei guasti RAM evidenti. Tuttavia non si riesce così ad individuare problemi di *sensibilità ai pattern.* La sensibilità ai pattern è una sorgente di guasti nei circuiti integrati ad alta densità. A causa dello schema geometrico del circuito alcune combinazioni di bit scritte in qualche istante nelle celle di memoria possono dar luogo ad innalzamenti od abbassamenti di bit in qualche altra posizione.

Questo problema può accadere nelle memorie RAM o nel microprocessore stesso. Quando questo problema sopravviene nel microprocessore è un guasto su cui l'utente non può fare nulla.

L'unica cosa che l'utente può fare è eseguire un pesante programma di prova rilasciato dal costruttore che sotto una specifica sequenza di istruzioni è stato dimostrato che causa un guasto analogo. Questo problema non sarà considerato, in quanto è ritenuto altamente infrequente una volta che il circuito lavora da piú di un anno. Tuttavia specialmente in caso delle memorie ad alta densità, la sensibilità ai pattern è un problema frequente che può essere diagnosticato facilmente usando una generazione algoritmica dei pattern.

Questo sistema sarà descritto nella prossima sezione.

Prove a pattern galoppanti

La prova a pattern galoppanti è usualmente abbreviata con la parola «galpat». Il principio di questa tecnica consiste nello scrivere valori binari successivi nelle celle di memoria e quindi compararli con gli altri della rimamente memoria prima di muoverli nella prossima locazione di memoria. In questa maniera se la scrittura nella cella di memoria zero influenza il contenuto della cella di memoria 102 questo

viene rilevato dalla prova.

In una tipica prova di galpat, la memoria viene inizializzata ad un valore conosciuto quali tutti uno o tutti zero.
L'algoritmo è il seguente:

1. Il contenuto della locazione L-1 viene confrontato con i contenuti di tutte le altre locazioni di memoria. Essi devono coincidere.
2. L'indirizzo L-1 è quindi incrementato di uno ed il passo precedente è ripetuto fin quando non siano state provate tutte le posizioni di memoria.
3. Il pattern iniziale è quindi complementato e si torna indietro al passo 1.

Sono possibili alcune variazioni a questo tipo di prova. Esse sono dette «zero ed uno a passo di marcia», «zero ed uno a passo d'uomo» e «pattern a passo di galoppo» (galpat uno e galpat due).

Teoricamente si dovrebbe scrivere tutti i pattern possibili in ciascuna locazione di memoria, verificare tutte le altre locazioni di memoria per controllare se qualcosa è cambiato.

In piú, dopo aver verificato ciascuna altra parola in memoria, si dovrebbe tornare indietro alla locazione di partenza per verificare che il suo pattern non sia stato modificato dalle prove in corso. Potrebbe infatti accadere che il fatto di provare tutte le altre locazioni di memoria potrebbe modificare il contenuto iniziale della cella, e quindi modificarlo ancora in modo che alla fine si ritrova allo stesso valore iniziale corretto.

Un possibile guasto non sarebbe individuato se non si torna indietro per verificare il contenuto della cella iniziale.

È facile vedere che questo tipo di prova richiede un numero elevatissimo di operazioni.

Per verificare con un semplice programma di prova 32K di memoria occorrono alcuni minuti.

Tale programma scriverà tutti «0» o tutti «1» oppure scriverà in ciascuna locazione di memoria il rispettivo indirizzo e quindi ruoterà questi indirizzi lungo la memoria disponibile.

Se si usa la tecnica galpat si può arrivare facilmente a mezza ora o anche a qualche ora.

Per questa ragione tali prove vengono eseguite soltanto durante le fasi iniziali di verifica del sistema o quando si sospetta qualche malfunzionamento. Non è praticamente pensabile il loro uso durante il funzionamento del microprocessore a meno di non usare una versione ridotta.

SIMULAZIONE ED EMULAZIONE

Introduciamo per prime le definizioni basilari.
La parola *Simulazione* si riferisce al fatto di rimpiazzàre funzionalmente i componenti hardware mediante un programma.

Si dice che in questo caso il componente hardware è simulato da software. Il programma genererà le stesse uscite sul componente hardware in risposta agli stessi ingressi. Sfortunatamente esso farà questa simulazione in maniera *piú lenta* del componente originale.

L'*Emulazione* si riferisce essenzialmente ad una simulazione effettuata in *tempo reale.*

In effetti numerosi emulatori simuleranno le operazioni di un sistema completo in maniera anche piú veloce del modello.

Per esempio molti sistemi bit-slice emulano le istruzioni di un altro computer. Essi eseguono le istruzioni del processor emulato alla stessa velocità o anche piú veloci.

La simulazione essenzialmente è usata per due tipi di apparecchiature: il microprocessore stesso e la memoria ROM.

La simulazione o l'emulazione della ROM è ottenuta eseguendo programmi in RAM come se fossero in ROM.

Questo è fatto normalmente durante la fase di sviluppo dei programmi. Chiaramente un programma inizialmente contiene un certo numero di errori e non deve essere messo in maniera finale nella ROM o nella PROM.

In un tipico sistema di sviluppo tale programma viene installato nella RAM e ivi provato e messo a punto. I due problemi principali sono il convertire gli indirizzi del programma finale in quelli richiesti dalla ROM e mantenere la compatibilità delle velocità.

Normalmente la RAM risiede ad uno specifico indirizzo fisico che non corrisponde all'indirizzo della ROM nel sistema finale.

Il secondo problema è un problema di sincronizzazione che c'è quando inizialmente è usata una RAM lenta ed il programma sarà poi installato su una ROM piú veloce.

Queste possibilità di emulazione e di simulazione della ROM fanno parte del sistema di sviluppo di ogni microcomputer e qui non verranno trattate in profondo dettaglio.

La simulazione o l'emulazione dello stesso microprocessore è piú complicata. Tale simulazione è usata in due casi:

1. quando la MPU stessa non è disponibile.
2. per facilitare in alcuni casi le prove di messa a punto.

Questi due casi non saranno visti in dettaglio.

Quando i programmi sono sviluppati in sistemi di notevoli dimensioni si usano *cross-programs.* Ad esempio un cross - assembler creerà codice 8080 in un IBM 370. È necessario provare la corretta funzionalità del codice 8080 risultante. Questo può essere fatto con un simulatore. Si userà perciò un simulatore che eseguirà tutte le istruzioni 8080 in un tempo simulato.

In questa maniera si prova la logica completa del sistema.

Una limitazione di questo simulatore è che non si può provare l' I/O fin quando non vengono passati, a tempo debito, i dati nelle locazioni di memoria predisposte.

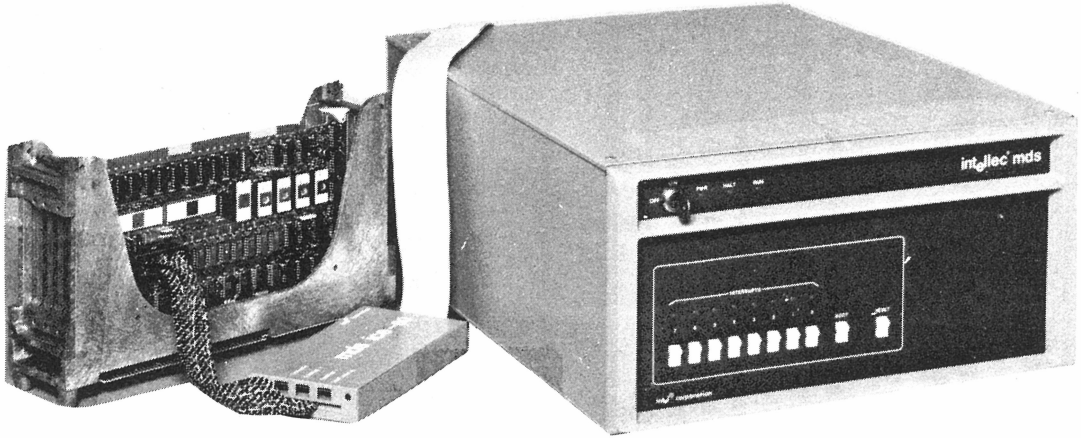


Fig. 8-19: Sviluppo software

Si deve quindi simulare i registri di input-output mediante locazioni di memoria. Sfortunatamente la temporizzazione dell'input-output è variabile spesso secondo leggi complesse.

Per questa ragione un simulatore è usato solo per provare la logica generale del programma. Ciò va bene per algoritmi numerici come la gestione di un floating-point, mentre è inadeguata per provare interfacce complesse di input-output.

In ogni sistema dove l'utente deve provare l'input-output in tempo reale, una delle possibilità è di emulare il microprocessore stesso.

Questo sistema è detto di «emulazione del circuito interno».

Emulazione del circuito interno

Tale tipo di emulazione è stato introdotto originariamente dalla INTEL nel suo sistema MOS ed adesso è disponibile nei principali sistemi di sviluppo di microprocessori anche come sistema indipendente.

La fotografia di un emulatore del circuito interno (ICE) compare in fig. 8-9. Una piastra speciale è stata inserita nel sistema INTEL MDS sulla sinistra e provvede a fornire le possibilità di emulazione.

Sulla destra compare il sistema in sviluppo. Lo stesso 8080 è stato rimosso ed al suo posto è stato collegato un «cavo ombelicale». Questo è il cavo che appare nell'illustrazione.

Tale cavo, a 40 fili, termina con un connettore a 40 piedini identico ad un 8080. La differenza è che tutti i segnali che passano su questo cavo sono generati o sono sotto il controllo di un emulatore di circuito interno invece che di un 8080 reale.

Quale è lo scopo di rimpiazzare un 8080 reale con un emulatore software? Le possibilità offerte da un emulatore sono di controllare e provare il sistema mediante consolle. È infatti possibile bloccare le operazioni dell'8080 ed esaminare i registri e cambiarli. Fare questo in un 8080 reale significherebbe aprirlo, piazzare delle microsonde con un microscopio per ottenere il contenuto dei registri.

In realtà non è possibile fare ciò in un 8080.

Usando un emulatore è possibile invece fermare le operazioni dell'8080 usando semplicemente degli alt di programma. Questa possibilità verrà spiegata in seguito. È possibile esaminare o cambiare i registri ed anche il contenuto della memoria. È possibile ancora eseguire da tastiera delle operazioni di input-output come la chiusura di un relay. È possibile inoltre fermare il microprocessore ed esaminare i registri o la memoria. Inoltre, tutte le operazioni possono essere eseguite con l'aiuto delle notevoli possibilità software messe a disposizione da un sistema di sviluppo.

Esaminare o cambiare la memoria può essere fatto in forma simbolica invece che in binario o in esadecimale. Questo modo di operare è detto «sistema di prova simbolico».

Per i punti di interruzione si può dare un indirizzo dove il programma si fermerà automaticamente. Quando, durante l'esecuzione, è raggiunta una locazione specifica, il microprocessore emulato si fermerà automaticamente e permetterà all'utente

di verificare registri e memorie.

Inoltre c'è la possibilità di avere una serie di fotografie della storia dei segnali sui bus durante uno specifico intervallo di tempo (trace - back).

Nel caso dell'INTEL ICE c'è la possibilità di avere questa storia per la durata di 44 cicli macchina.

Ogni volta che viene incontrato un punto di interruzione, l'emulatore ferma l'esecuzione e fornisce all'utente la possibilità di correzioni simboliche.

Tipicamente quando si incontra un errore al punto di interruzione questo non è dovuto all'ultima istruzione eseguita, ma è causato da una istruzione precedente del programma.

Il problema è di localizzare tale istruzione, ossia è un problema di trace - back. Quando esiste questa possibilità, è possibile esaminare i segnali precedenti e determinare quale istruzione fosse stata eseguita prima dell'errore. Se la possibilità di trace-back non è sufficiente, si può mettere un punto di interruzione prima e si ottiene così una ulteriore storia del sistema.

Questo processo può andare avanti finchè non si trova l'errore. Tale emulatore non richiede, per essere usato, delle configurazioni hardware o software notevoli. Esso rappresenta una possibilità di ricerca errori e quando si pone l'esigenza di avere un dispositivo che verifichi il funzionamento di un sistema comprese le piastre di input - output o le interfacce, allora tale emulatore è essenziale.

CONTROLLO PASSO PASSO (DEBUGGING) DI UN PROCESSORE TEORICO

Dopo che è stato verificato che tutti i livelli logici sono corretti, il sistema è pronto per alcuni *semplici* programmi di test. È buona norma non essere troppo sicuro di sé in questa fase! Si proceda a piccoli passi, come ad esempio: si indirizzi sequenzialmente ogni possibile posizione di memoria, si salti a «0000» esadecimale continuamente, si entri da una porta, e si pongano in uscita i dati presentati in ingresso. I test devono essere allocati in PROM separate in modo che possano essere eseguite individualmente.

Il test degli indirizzi deve essere realizzato commutando ciascun bit delle vie di indirizzo ad intervalli via via piú lunghi con onde quadre.

Il test di salto è così corto, che è spesso possibile osservare dinamicamente con un oscilloscopio tutte le vie in ciascuna condizione. Inoltre, tutti i bit di indirizzo, da A2 ad A15, devono essere nel test proposto tutti zero. L'ingresso-uscita permette che ciascun bit di ingresso sia controllato. Se un bit è mantenuto alto, il bit di uscita corrispondente deve anche presentarsi a livello alto quando abilitato. Se non lo è, c'è un errore nello schema di ingresso - uscita del sistema, o nel microprocessore.

Adesso il test diventa interessante. Si provino programmi piú lunghi, procedendo fino a verificare l'intero programma applicativo. *A tal punto, tutti i problemi possono essere soltanto di tipo software.* Se si è sicuri che il problema è di tipo hardware,

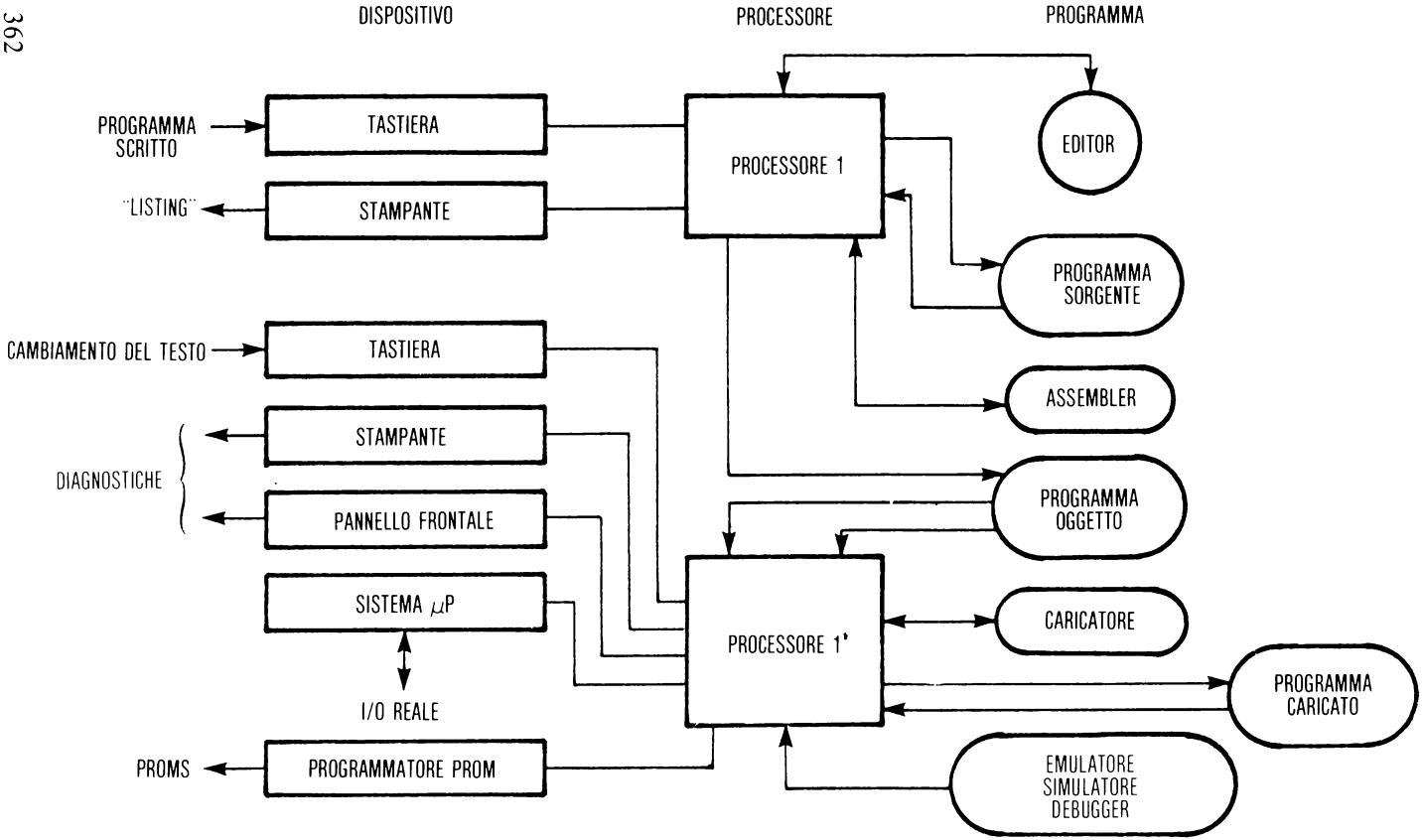


Fig. 8-20: Diagramma di flusso di «debug»

si torni indietro e si scrivano altri semplici programmi di test per verificarlo si ricordi che se un limitato numero di istruzioni opera correttamente, normalmente tutto il sistema è corretto.

È di aiuto sapere che sono disponibili ROM codificate per un controllo passo passo dei sistemi e per gran parte degli sviluppi di prototipi. Essi sono normalmente chiamati programmi di test e verificatori passo esadecimali (o ottali), o ancora «System Monitors». La figura 8-20 mostra il flusso di controllo passo passo per una situazione tipica.

Problemi tipici specifici per i micro

Segue una lista di alcuni problemi che l'autore ha trovato interessanti:

- Un bit di indirizzo errato sul microprocessore faceva si che ogni programma codificato oltre «1FFF» in esadecimale non fosse eseguito correttamente.
- Una EPROM con leakage eccessivo aveva perduto i suoi dati prima che potesse essere utilizzata nel sistema dopo la sua programmazione.
- I circuiti PMOS e NMOS non possono mai essere connessi senza buffer. Ciò è vero per tutte le famiglie. «Compatibile con TTL» significa che può essere connesso con TTL — *non che esso può essere connesso con ogni altra cosa indicata compatibile con TTL*!
  Ciò può causare seri problemi. Ad esempio: una via di indirizzo PMOS verso una RAM NMOS può causare che un bit non individuabile della RAM si perda! Questi problemi sono inoltre sensibili al calore e alla alimentazione. Il sistema da usare può lavorare in uno specifico campo di temperatura e di alimentazione. Si controlli che le specifiche siano adatte.
- Le RAM dinamiche possono guastarsi in una singola posizione di bit *casuale*. Questo spiega la presenza di rivelazione di errore, parità e correzione di errore in sistemi con molta memoria.
- I bus: come regola, non si connetta piú di un ingresso e una uscita a qualunque via del bus. Trascurare questa regola comporta una maggiore sensibilità al rumore a causa del sovraccarico. La via che piú comunemente viola tale regola è il RESET.
- *Non si devono avere dubbi sul livello di ciascuna terminazione. Occorre conoscere ciò che è a livello basso e ciò che è a livello alto.* Se si hanno dubbi, si misuri il circuito sullo zoccolo, e si chiami il costruttore per individuare la prima terminazione dell'integrato e i livelli corretti.

In figura 8-21 è indicata una sequenza dei possibili malfunzionamenti.

UN BIT SU 16.384

Il progetto di un multiplatore indicato nel capitolo 8 richiede 6 mesi-uomo per essere controllato passo passo completamente, con due ingegneri impegnati a tempo pieno per quel lavoro, disponendo di tutti gli strumenti indicati in quel capitolo.

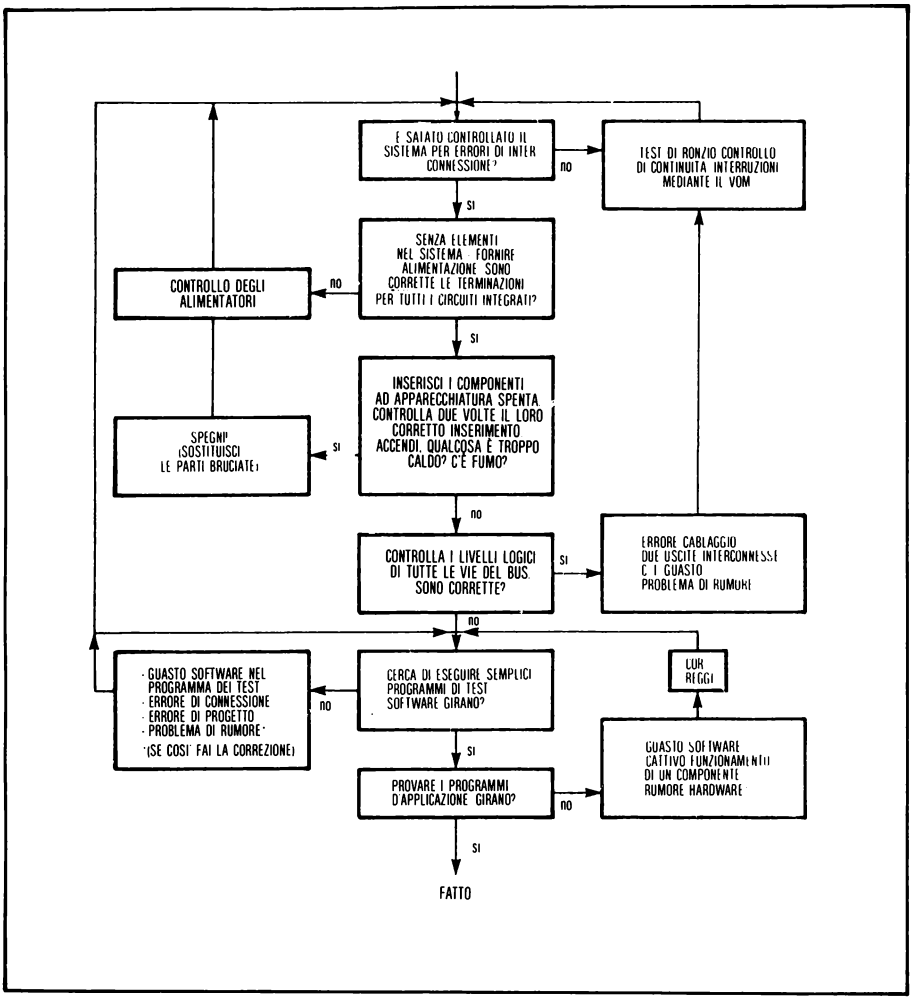


Fig. 8-21: Diagramma di flusso di errata funzionalità

Pertanto, il costo di controllare passo passo quel sistema costa:

– salario di 6 mesi:	$ 10.000
– apparecchiature per 6 mesi:	$ 15.000 (se affittati)
	$ 8.000 (usati per 5 anni)

Questo paragrafo mette a fuoco i problemi reali, come essi si sono presentati.

Prima settimana

Completamento della versione cablata. Inizio della fase di test.

Seconda settimana

Completamento di una fase di test grossolano. Ciascun modulo presenta circa 20 errori su 1000 connessioni. Si é verificato un cortocircuito tra la alimentazione e la massa per una piastra. Erogando una corrente elevata alla piastra senza componenti si é bruciato il cortocircuito e si é scoperta una capacità cortocircuitata nella piastra di memoria.

Terza settimana

Ciascuna piastra é stata provata nei suoi segnali logici, separatamente. Più di metà degli errori per piastra era dovuta alle connessioni. Le piastre di circuito stampato erano realizzate con moduli «wire-wrapped».

Quarta settimana

Sul prototipo di sistema sono state eseguite tutte le semplici routine di test. Sono state individuate celle di memoria guaste nelle piastre RAM mediante un programma di test di memoria che scriveva uni e zeri in ogni cella.

Quinta settimana

Programma di sistema con problemi di carico del bus. È stata aggiunta una E-PROM nella piastra CPU. Le operazioni di ingresso-uscita di programma operano correttamente.

Sesta settimana

Solo problemi software sono ora esaminati. Le interconnessioni tra i circuiti stampati sono pronte per il controllo prima di realizzare le piastre.

Settima settimana

Interconnessioni di piastra approvate, circa 5 errori per piastra. Il sistema ha problemi di deriva: funziona per alcune ore e dopo dà segnali qualsiasi al minicomputer.

Ottava settimana

Restituzione delle piastre e controllo passo passo. Sostituzione delle piastre cablate con altre stampate, una alla volta, per trovare gli errori.

Nona settimana

Nuovamente errori di interconnessioni nelle piastre. Il sistema lavora ancora in

modo incorretto. È stato usato intensivamente un Analizzatore Logico per individuare il problema.

Decima settimana

È stato trovato un driver di bus guasto nella piastra USART verso il mini. I guasti si verificano circa uno al giorno. Le piastre stampate sono state completate. Qualche volta il sistema riceve dati errati dal terminale. È usato un emulatore aderente al circuito per verificare la routine di prelievo dati. È usata una tecnica di traccia di ritorno. Gli errori si verificano circa ogni 8 ore e quindi difficilmente sincronizzabili.

Undicesima settimana

Discussioni tra programmatori e progettisti-sessione di esame strutture e circuiti. Il guasto é individuato venerdì. Due problemi.

Dodicesima settimana

Era usato un bit errato nella EPROM per il programma e il bit di carry non era posto a zero entrando nella routine di interruzione. Era usata una somma con carry, invece che una senza carry. La istruzione relativa determinava la posizione del dato da trasmettere e pertanto occasionalmente veniva prelevato il dato non corretto, quando si verificava un riporto dopo una interruzione. Il bit errato é stato individuato controllando la PROM rispetto alle quattro fasi di listing. La istruzione errata é stata individuata usando l'Analizzatore Logico con il ritorno di traccia, quando é stata eseguita una lettura dalla posizione incorretta.

Fasi successive

Dopo la fine della dodicesima settimana sono stati utilizzati tre sistemi identici, per disporre di una statistica degli errori. Ci furono meno errori nei multiplatori, con intervalli di fuori uso dieci volte inferiori, che nei minicomputer ai quali essi erano connessi.

CONCLUSIONI

Quando si verifica un problema i componenti, i software e il rumore sono i soli oggetti da esaminare. I diagrammi di flusso presentati hanno indicato un semplice metodo di approcci dei problemi connessi ai sistemi a microprocessore. Sono state indicate le apparecchiature necessarie per un corretto controllo passo passo del microprocessore, e sono stati sviluppati esempi. In figura 8-22 sono indicate tutte le apparecchiature necessarie per un sistema prototipo. È da tener presente il costo: tipicamente 45.000 $. Si provi ad usare meno apparecchiature e ci si accorgerà che il tempo necessario per individuare il guasto aumenta.

Futuri mezzi hardware di controllo passo passo saranno orientati verso l'analisi degli stati. Caratteristiche di tali macchine saranno un gran numero di stati, tracce e capacità di trigger, oltre alla abilità di dare opportune rappresentazioni sullo schermo degli stati, in qualsiasi forma mnemonica. Inoltre il loro uso nei minicalcolatori e nei grossi elaboratori diventerà sempre più esteso e alcuni sistemi conterranno un analizzatore per realizzare l'autodiagnostica.

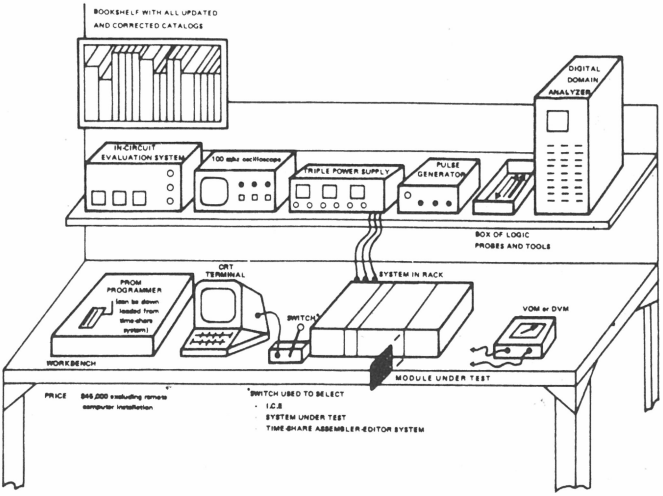


Fig. 8-22: Apparecchiatura prototipo

RICORDA!
NON....

Non si usi un circuito integrato se non si é connesso a massa, o se l'umidità ambiente é elevata. Una carica statica, come quella generata camminando su un tappeto in un giorno secco distrugge gran parte dei circuiti MOS con una scarica di centinaia di volt.

Non si installi una piastra nel calcolatore a meno che:

- la alimentazione é spenta
- non si sia atteso 15 secondi (tutte le cariche sono state annullate in dieci secondi).

Non si accenda e spenga il calcolatore con il dischetto inserito: transitori lo potrebbero cancellare. Si accenda prima il calcolatore, si agisce come é necessario con il dischetto, e si spenga poi eventualmente il calcolatore.

Non si selezionino più moduli in ingresso generando indirizzi non corretti, altrimenti si deve disporre di una MPU di riserva.

Non si superino i livelli di alimentazione consentiti. Si controllino.

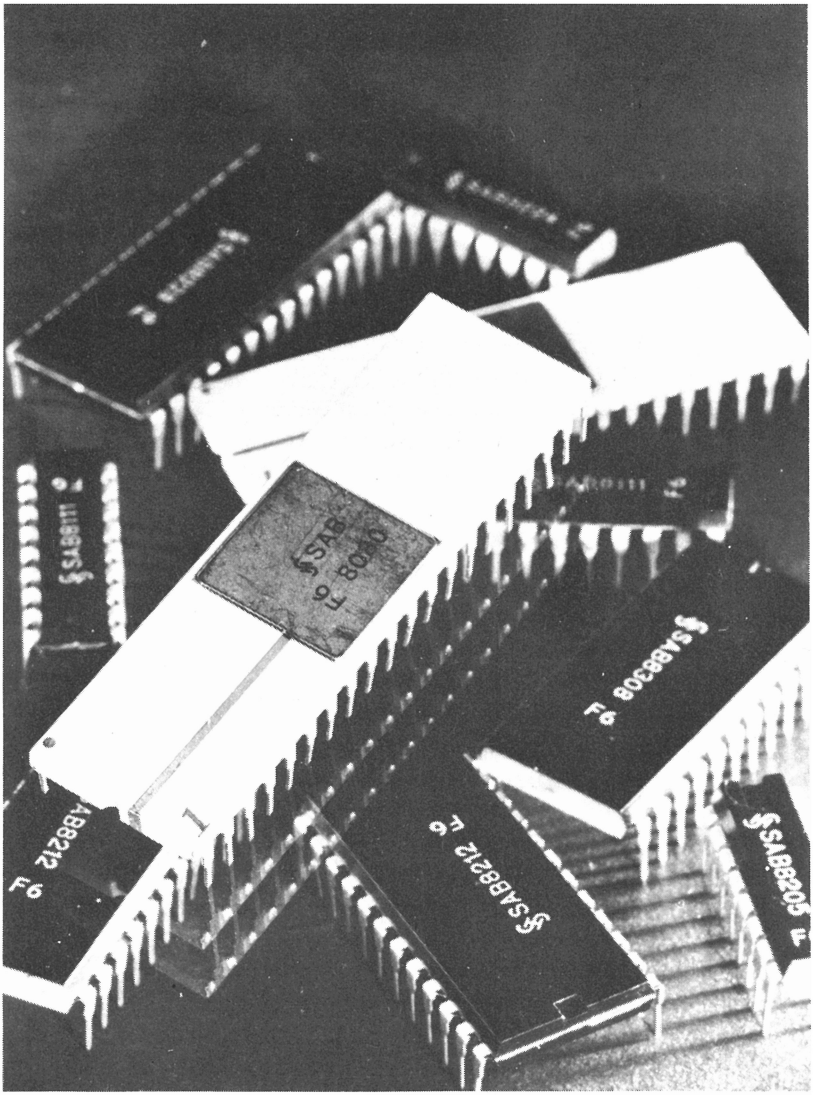


Fig. 8-23: Integrati della famiglia 8080

CAPITOLO 9

EVOLUZIONE

EVOLUZIONE TECNOLOGICA

Iniziando con i fondamenti della interconnessione dei sistemi, si é proceduto attraverso le tecniche di interfacciamento. Nel frattempo, la evoluzione é stata verso l'uso di interfacce completamente integrate. Il telaio precedentemente usato é stato rimpiazzato da un ridotto numero di circuiti integrati LSI in un contenitore di dimensioni ridotte. Il futuro sviluppo di circuiti periferici più intelligenti aumenterà la flessibilità e le capacità.

Il processore centrale che é alla base di ogni sistema é già contenuto in un singolo modulo. Il futuro *microcalcolatore in singolo modulo* eliminerà i problemi di interconnessione delle memorie e del processore. Questi moduli in un unico integrato conterranno ROM, RAM e capacità di I/O tali da realizzare gran parte dei compiti di interfaccia. Tali componenti sono già stati presentati: il 9940 della Texas Instrument, l'Intel 8048, il 3870 della Fairchild-Mostek. Essi contengono da 1 a 2 K di ROM, da 64 a 128 parole di RAM, il clock, il timer e la MPU. Le 16 terminazioni rese disponibili dalla assenza degli indirizzi mediante bus permettono la presenza di due porte I/O di 8 bit.

Il 9940 della Texas Instrument é un microcalcolatore a 16 bit con 1 K di ROM, RAM e ingressi-uscite in unico modulo. La potenza di istruzioni a 16 bit permette la realizzazione di un set completo di istruzioni, includendo la divisione e la moltiplicazione in hardware. Sfortunatamente, é una limitazione la limitata capacità della ROM.

L'Intel 8048 integra una PROM di 1K x 8, un registro di 32 byte e permette 27 diversi I/O su relative vie. Una EPROM, la 8748, permette che il programma sia cancellato e riprogrammato durante lo sviluppo. La versatilità ottenuta usando una ROM cancellabile, nello stesso modulo del processore e dell'I/O, rende l'8748 facilmente adattabile alle esigenze di interfaccia che evolvono.

Infatti, la prima interfaccia di tale tipo é ora disponibile dalla Intel. Chiamato UPI o 8041, esso é un microelaboratore completo in un singolo modulo, dedicato al controllo di una singola periferica. Poiché esso può essere programmato costituisce una interfaccia programmabile di uso generale che può connettere l'8080, l'8085, o l'8048 fino a qualunque altra periferica che richieda velocità di controllo minore di 20.000 byte al secondo. Il 3870 della Mostek-Fairchild integra una ROM di 2K, una RAM ed è compatibile in software con lo F8.

INTERFACCE PROGRAMMABILI

A causa del basso costo dei microprocessori in singolo modulo, i moduli di interfaccia di componenti stanno diventando «intelligenti», cioé equipaggiati con processore. Essi ricevono istruzioni dalla MPU, e realizzano tutti i controlli e le sequenze necessari. Un microprogramma interno al modulo realizza tutte le decodifiche e le sequenze richieste.

È interessante notare che la complessità di una MPU standard é di circa 6000 transistori. La complessità di un FDC o di un CRTC é tra 15.000 e 22.000 transistori.

Le interfacce in singolo modulo sono processori per applicazioni specifiche per il controllo di componenti.

Al crescere del livello d'integrazione, l'intero controllore sarà eventualmente compreso in un singolo modulo.

COSTI

Il costo delle interfacce rimarrà probabilmente più alto del costo dei processori, a causa della maggiore complessità e il minor volume. Tuttavia, esso é già diventato abbastanza trascurabile rispetto al costo delle periferiche.

«SOFTWARE PLASTICO»

Man mano che un algoritmo software diventa ben definito, esso può essere consolidato in una LSI a basso costo. Ciò é il «software plastico»: programmi che possono essere acquistati come un modulo LSI.

In un passo successivo della evoluzione, é probabile che gran parte degli algoritmi o programmi che sono stati indicati nel volume saranno sviluppati come parti di un modulo LSI complesso. Essi diventeranno software plastico.

L'interfacciamento diventerà essenzialmente una semplice interconnessione dei moduli necessari. Quando ciò avverrà, é sperabile che le tecniche qui presentate contribuiscano alla loro comprensione.

In conclusione, sono state presentate passo passo diverse tecniche. Partendo dall'interfacciamento della CPU alla RAM, alla ROM e ai moduli di I/O, fino alle interfacce intelligenti del disco floppy. Sono stati anche coperti altri soggetti come l'alimentazione e la conversione analogico-digitale. Questi argomenti sono necessari per avere una capacità completa di operare impiegando microprocessori digitali. La interfaccia va dalla presa di alimentazione alla più remota periferica. Man mano che la tecnologia progredisce il progettista di interfacce deve progredire verso una visione più ampia dei problemi, acquisendo esperienze nella alimentazione e nel progetto analogico. In tali condizioni si é capaci di interfacciare unità di qualsiasi tipo.

GLI AUTORI

AUSTIN LESEA *ha sviluppato un prodotto microprocessore per ricerca, educazione e applicazioni industriali. Progettista del microprocessore per applicazioni educative lavora nella Università della California, a Berkeley, dove ha presentato corsi e seminari nel campo dei microprocessori a circa 1000 partecipanti, in tutto il mondo.*

Il Dr. RODNAY ZAKS *ha pubblicato più di 30 libri e contributi di ricerca sui microprocessori oltre ad avere sviluppato diversi prodotti basati su microprocessori, per applicazioni di ricerca e industriali. Egli ha presentato corsi e seminari per più di 3000 partecipanti, in tutto il mondo, da livello introduttivo alle tecniche di microprogrammazione a «bit-slice».*

Questo libro é il risultato della esperienza di progetto ed educativa degli autori.

APPENDICE A

PRODUTTORI DI MICROPROCESSORI

AMD (Advanced Micro Devices)
901 Thompson Place
Sunnyvale, CA 94608
(408) 732-2400
Telex: 346306

AMI (American Microsystems)
3800 Homestead Road
Santa Clara, CA 95051
(408) 246-0330

Data General
Southboro, MASS 01772
(617) 485-9100
Telex: 48460

Electronic Arrays
550 East Middlefiedl Road
Mountain View, CA 94043
(415) 964-4321

EMM Semiconductor
3216 West El Segundo Blvd.
Hawthorne, CA 90250
(213) 675-9141

Fairchild Semiconductor
1725 Technology Drive
San Jose, CA 95110
(408) 998-0123

GI (General Instruments)
600 West John Street
Hicksville, NY 16002
(516) 733-3107
TWX: (510) 221-1666

Harris Semiconductor
Box 883
Melbourne, FL 32901
TWX: (510) 959-6259

Hitachi
2700 River Road
Des Plaines, IL 60018
(312) 298-0840

Hughes Microelectronics
500 Superior Avenue
Newport Beach, CA 92663
(714) 548-0671

Intel
3065 Bowers Avenue
Santa Clara, CA 95051
(408) 246-7501
Telex: 346372

Intersil
10090 North Tantau Avenue
Cupertino, CA 95014
(408) 996-5000
TWX: (916) 338-0228

ITT Semiconductors
74 Commerce Way
Woburn, MASS 01801
(617) 935-7910

MMI (Monolithic Memories)
1165 East Arques Avenue
Sunnyvale, CA 04086
(408) 739-3535

MOS Technology
950 Rittenhouse Road
Norristown, PA 19401
(215) 666-7950
TWX: (510) 660-4033

Mostek
1215 West Crosby Road
Carollton, TX 75006
(214) 242-0444
Telex: 30423

SGS-ATES
79 Massosoit Street
Waltham, MASS 02154
(617) 891-3710
Telex: 923495WHA

Sharp
10 Keystone Place
Paramus, NJ 07652
(201) 265-5600
Telex: 134327

Siemens
3700 East Thomas
Box 1390
Scottsdale, AZ 85252
(602) 947-2231

Signetics
811 East Arques Avenue
Sunnyvale, CA 94086
(408) 739-7700

Solid State Scientific
Montgomeryvale, PA 18936
(215) 855-8400
Telex: (510) 661-7267

Synertek
3050 Coronado Drive
Santa Clara, CA 95051
(408) 984-9800
TWX: (910) 338-0135

TI (Texas Instruments)
Digital Systems Division
P.O. Box 1444
Houston, TX 77001
(713) 494-5115

Thomson-CSF, Sescosem
50 Rue JP Timbaud
Courbevoie 92, France
788-50-01
Telex: 610-560

Western Digital Corp.
3128 Redhill Avenue
Newport Beach, CA 92663
(714) 557-3550
TWX: (910) 595-1139

Zilog
10460 Bubb Road
Cupertino, CA 95014
(408) 446-4666
TWX: 910-338-7621

APPENDICE B

PRODUTTORI DELL'S100

SISTEMI CALCOLATORI	COUT USA ($)
Byte Shop Byt-B	349.00
Computer Power & Light COMPAL-80 (assembled)	2,300.00
Cromemco Z-1 (assembled)	2,495.00
Cromemco Z-2K	595.00
Electronic Control Technology ECT-100-8080	320.00
Electronic Control Technology ECT-100-Z80	420.00
Equinox 100	699.00
Forethought Products KIMSI connector and KIM (6502)	370.00
IMSAI 8080 Computer (chassis, power, & CPU)	699.00
IMSAI PKG-1	4,444.00
IMSAI PKG-2	9,013.00
MITS Altair 8800B	875.00
Morrow's Micro Stuff Signa 100	250.00
PolyMorphic Systems POLY-88 System 0	525.00
PolyMorphic Systems POLY-88 System 2	735.00
PolyMorphic Systems POLY-88 System 6	1,575.00
PolyMorphic Disk System (1 disk)	3,250.00
Processor Technology SOL-PC Single Board	475.00
Processor Technology SOL-10 Terminal Computer	795.00
Processor Technology SOL-20 Terminal Computer	995.00
Processor Technology System I	1,649.00
Processor Technology System II	1,883.00
Processor Technology System III	4,237.00
Quay AI Z-80 CPU, SIO, PIO, ROM, Programmer Board	450.00
Technical Design Labs XITAN Alpha 1	769.00
Technical Design Labs XITAN Alpha 2	1,369.00
Vector Graphics Vector I	699.00
Vector Graphics Vector I without PROM/RAM	519.00
Vector Graphics Vector I without CPU	499.00
Vector Graphics Vector I without CPU, PROM/RAM	349.00
Western Data Systems DATA HANDLER (used MOS 6502)	179.95
Western Data Systems DATA HANDLER (bare bones)	79.95

PIASTRE CPU SUPPLEMENTARI

Affordable Computer Products AZPU (uses Z-80)	249.00
Alpha Micro Systems AM-100 (16 bit)	1,495.00
CGRS 6502	?
Cromemco ZPU (uses Z-80/4 microprocessor)	295.00
IMSAI MPU-A (requires additional boards)	190.00
MRS AM6800 CK (uses 6800 MPU)	110.00
MRS AM6800 (without the 6800 MPU chip)	78.00
MRS AM6800 PC Board	30.00
R.H.S. Marketing Piggy-Back Z80-80 (assembled)	159.95
SD Sales Z-80 CPU	149.00
Technical Design Labs Z-80 (uses Z-80)	269.00

PIASTRE MEMORIA DI LETTURA/SCRITTURA

Advanced Microcomputer Products Logos 8K RAM	219.95
Advanced Microcomputer Products 801C 8K RAM	207.95
Advanced Microcomputer Products 32K RAM	1,150.00
Artec 32K Memory Board (8K, 250 nS)	290.00
Artec 32K Memory Board (32K, 250 nS)	1,055.00
Associated Electronics 15K Pseudo-Static	349.95
Base-2 BKS-A	98.00
Base-2 BKS-B (450 nS)	123.00
Base-2 BKS-Z	143.00
BISI CCK Board (64K)	190.00
Crestline Micro Systems (8K, low power, assembled)	179.00
Cromemco 4KZ (4K 4MHz) (Bank selectable)	195.00
Cromemco 16KZ (16K 250 nX access and cycle)	495.00
Cybercom MB6A Blue Board (8K static)	250.00
Cybercom MB7 (16K low power static)	525.00
Data Sync 16K (assembled)	298.00
Duston 8K Memory Board (bare)	29.00
Dutronics 4KLST (4K low power static)	139.00
Dutronics 8KLST (8K low power static)	285.00
E.E. & P.S. 8K (8K static)	295.00
E.E. & P.S. 16K (16K dynamic)	599.00
E.E. & P.S. 32K (32K dynamic)	895.00
Electronic Control Technology 8KM (8K 215 nS)	295.00
Electronic Control Technology 16K RAM (16K static)	555.00
Electronic Control Technology 16K RAM (with only 4K)	169.00
Electronic Control Technology 16K RAM (with only 8K)	295.00
Electronic Control Technology 16K RAM (with only 12K)	425.00
Extensys RM64-32 (32K)	895.00
Extensys RM64-48 (48K)	1,195.00
Extensys RM64-64 (64K)	1,495.00
Franklin Electric 8K Static RAM	225.00
Godbout Econoram (4K static)	99.95

Godbout Econoram II (8K)	163.84
IMSAI RAM 4A-4 (4K without sockets)	139.00
IMSAI RAM 4A-4 (4K with sockets)	159.00
IMSAI 65K (dynamic)	2,599.00
IMSAI 32K (dynamic)	749.00
IMSAI 16K (dynamic)	449.00
Kent-Moore 4K (assembled)	107.00
Microdesign MR8 (EPROM/RAM)	124.95
Micromation JUMP START (4K static)	145.00
Midwest Scientific Instruments PROM/RAM Board	95.00
Mikro-D MD-2046-4 (4K static)	205.00
Mikro-D MD-2046-8 (8K static)	345.00
Mikro-D MD-2046-12 (12K static)	485.00
Mikro-D MD 2046-16 (16K static)	625.00
MiniMicroMart C-80-4K-100 (4K blank board)	39.95
MiniMicroMart C-80-4K-700 (4K blank board plus)	49.95
MiniMicroMart C-80-4K-300S (4K 2102)	79.95
MiniMicroMart C-80-4K-300LP (4K 9IL02A)	99.95
MiniMicroMart C-80-4K-350LP (4K 9IL02C)	129.95
MiniMicroMart C-80-16K-300 (16K E MM4200)	479.95
MITS 88-4MCS (4K static)	167.00
MITS 88-16MCS (16K static)	765.00
MITS 88-S4K (4K dynamic)	155.00
Morrow Intelligent Cassette (512 static)	96.00
Mountain Hardware PROROM (256)	164.00
Omni (16K static)	459.00
Omni with paging option (16K static)	468.00
Prime Rodi x 40K (dynamic)	1,490.00
Prime Rodi x 48K (dynamic)	1,580.00
Prime Rodi x 56K (dynamic)	1,670.00
Prime Rodi x 64K (dynamic)	1,750.00
Processor Technology 4KRA (4K static with sockets)	154.00
Processor Technology 8KRA (8K static with sockets)	295.00
Processor Technology 16KRA (16K static assembled)	529.00
PolyMorphic Systems MEM-8K (8K static)	300.00
R.H.S. Marketing DYNABYTE 16K (dynamic, a mbled)	485.00
J-K Electronics DYNA-RAM 16 (16K)	339.00
S. D. Sales Company 4K (4K static)	89.95
Seals Electronics 8KSC-8 (8K static)	269.00
Seals Electronics 8KSC-Z (8K 250 nS)	295.00
Seals Electronics 8KSCL M (less memory chips)	124.00
Seals Electronics 16KSC-16 (16K static)	579.00
Solid State Music M8-4 (4K 9IL02A)	129.95
Solid State Music M8-4 (8K 9IL02A)	209.00
Solid State Music M8-4 (board only)	30.00
Solid State Music M8-4 (board only)	35.00
Solid State Music M8-6 (8K 9IL02APC static)	265.00
Solid State Music M8-7 (16K static)	525.00
Technical Design Labs Z8K (4K 215 nS)	169.00
Technical Design Labs Z8K (8K 215 nS)	295.00

Technical Design Labs Z12K (12K 215 nS)	435.00
Technical Design Labs Z16K (16K 215 nS)	574.00
Technical Design Labs Z Monitor Board with 2K RAM	295.00
Vandenberg 16K RAM (dynamic)	299.00
Vector Graphics 8K RAM	265.00
Vector Graphic Reset and Go PROM/RAM	89.00
Xybek PRAMMER (256 bytes & 1702 PROMs)	189.00

PIASTRE DI PROGRAMMATORE PROM

Cromemco BYTESAVER for 2704 & 2708	145.00
Mountain Hardware PROROM (AMI 6834)	164.00
Quay AI Z-80 with 2708 Programmer	450.00
Szerlip Enterprises The Prom Setter (1702A and 2708)	165.00
Xebek PRAMMER for 1702 (with 1702 & RAM)	209.00

PIASTRE SOFTWARE A INNESTO

Computer Kits Power-Start	165.00
Cromemco Z80 Monitor Board with PROM Programmer	220.00
Godbout 8080 Software Board	189.95
Microdesign MR8 with MM2K	224.45
Micronics Better Bug Trap (assembled)	180.00
Midwest Scientific Instruments PROM/RAM Monitor	245.00
Mountain Hardware PROROM	164.00
National Multiplex Corp No. 2 SIO with monitor	140.00
Processor Technology ALS-8 (assembled)	425.00
Processor Technology ALS-8 with SIM-I	520.00
Processor Technology ALS-8 with TXT-I	520.00
Technical Design Labs Z System Monitor Board	295.00
Vector Graphics Reset and Go (2 1702A)	129.00
Vectur Graphics Reset and Go (3 1702A)	159.00

PIASTRE DI INTERFACCIA SERIALE

Advanced Microcomputer Products (3P + S compatible)	125.00
Cromemco TU-ART (2 parts)	195.00
IMSAI SIO 2-1 (one part, without cables)	125.00
IMSAI SIO 2-2 (two parts, without cables)	156.00
IMSAI SIO (serial, parallel, & tape interface)	195.00
Morrow Intelligent Cassette with one part	103.00
MiniMicroMart C80-SI/O-300 (TTL)	44.95
MITS 88-2SIO (one part)	150.00
MITS 88-2SIO + SP (two parts)	188.00
MITS 88 SIOB	124.00
National Multiplex Corp No. 2 SIO with ROM	140.00

Processor Technology 3P+S (with sockets)	149.00
Solid State Music I/O-2 (two parts)	47.50
Solid State Music I/O-2 (PC board only)	25.00
Technical Design Labs Z Monitor Board (two parts)	295.00
WIZARD PSIOB (3P+S compatible)	125.00

PIASTRE DI INTERFACCIA ANALOGICA

Cromemco D+7AIO (7analog inputs & 7 outputs)	145.00
Micro Data ADC/DAC	250.00
MITS 88-ADC (assembled only)	524.00
MITS 88-Mux (assembled only)	319.00
MITS AD/DA (assembled)	235.00
PolyMorphic Systems ADA/I (1 analog output)	145.00
PolyMorphic Systems ADA/2 (2 analog outputs)	195.00

PIASTRE MODEM

International Data Systems 88-MODE M	199.00
Hayes 80-103A (assembled)	279.95
Hayes 80-103A (board only)	49.95

PIASTRE DI INTERFACCIA DI CASSETTE AUDIO

Affordable Computer Products Triple Standard	135.00
DAJEN Cassette Interface	120.00
DAJEN Universal Cassette Interface (Relay Control)	135.00
IMSAI MIO (tape interface, parallel, & serial)	195.00
MiniTerm Associates MERLIN with cassette interface	298.00
MITS 88-ACR	145.00
National Multiplex Corp No. 2 SIO with ROM	140.00
Morrow Intelligent Cassette Interface	96.00
Morrow Intelligent Cassette Interface (3 drives)	102.00
PerCom Data CI-812	89.95
Processor Technology CUTS	87.00
RO-CHE with Tarbell (two parts)	245.00
RO-CHE with Tarbell (four parts)	245.00
Tarbell	120.00

PIASTRE DI INTERFACCIA DI UNITÀ A NASTRO

MECA ALPHA-I System	400.00
Micro Design Model 100 (assembled)	600.00
Micro Design Model 200 (assembled)	875.00
MicroLogic M712 DG PhiDeck	69.95
National M.C. 2 SIO (R) 1 ROM	169.95

National M.C. 2 SIO (R) 2 ROM	189.95
National M.C. 2 SIO (R) with 3M3 (3M drive)	369.90
National M.C. 2 SIO (R) with 3M3 (mini 3M drive)	339.90

PIASTRE DI INTERFACCIA DI DISCO FLOPPY

Alpha Micro Systems AM-200 Controller	695.00
Alpha Micro Systems AM-201 Controller	695.00
CHP Floppy Disk Controller	300.00
Computer Hobbyist Products Controller	300.00
Computer Hobbyist Products (single drive)	850.00
DigiComm 8040 Floppy Disk Controller	265.00
Digital Systems IBM campatible	1,595.00
Digital Systems dual IBM campatible	2,170.00
iCOM Microfloppy Model FD2411 (assembled)	1,095.00
IMSAI F IF	599.00
IMSAI F DC2-1 & F IF	1,694.00
IMSAI F DC2-2 & F IF	2,789.00
INFO 2000 Adapter (without RAM)	120.00
INFO 2000 Adapter (with 4K RAM)	160.00
INFO 2000 Adapter - Per Sci 1070 Controller	860.00
Micromation Universal Disc Controller	229.00
Micromation MACRO DISC System, Model 164K	900.00
Micromation MACRO DISC System, Model 256K	1,100.00
Micropolis 1053 Mod II (630K)	1,795.00
Micropolis 1043 Mod II (315K)	1,095.00
Micropolis 1053 Mod I (286K)	1,545.00
Micropolis 1043 Mod I (143K)	945.00
MITS 88-DCDD (Controller & disk)	1,425.00
MITS 88-DISK	1,215.00
North Star Computers MICRO-DISK	699.00
PerCom Data Co.	695.00
Peripheral Vision interface and floppy	750.00
Peripheral Vision IFF-KC interface	245.00
Pertec RD2411	1,095.00
Processor Applications FDC-1016K Controller	395.00
Processor Technology Helios (dual)	1,895.00
Realistic Controls ZI/25	1,095.00
Synetic Designs interface and floppys	2,690.00
Tarbell Bare Board Interface	40.00
Tarbell Interface	190.00

PIASTRE DI INTERFACCIA DI DISCO HARD

IMSAI DISK-50	12,500.00
IMSAI DISK-80	14,700.00
IMSAI DISK-200	24,500.00
IMSAI Interface (assembled)	3,900.00

PIASTRE PROM

Crea Comp M 100/16 (16K, 2116)	485.00
Crea Comp M 100/16 (with parity)	560.00
Crea Comp M 100/32 (32K, 2116)	885.00
Crea Comp M 100/32 (with parity)	990.00
Cromemco BYTESAVER (8K)	145.00
Cromemco 16KPR-K (16K, Bank selectable)	145.00
DigiComm Byteuser (uses 2708)	65.00
Digiteck PROM CARD (2K assembled without PROMS)	56.95
Electronic Control Technology 2K ROM/2K RAM	120.00
Godbout Econoram (2K)	135.00
Godbout Econoram (4K)	179.95
Godbout Econoram (8K)	269.00
IBEX 16K PROM Board	85.00
IMSAI PROM 4-4 (4K PROM)	399.00
IMSAI PROM 4-512 (I/2K PROM)	165.00
Microdesign MR8 (for 2708)	99.50
Midwest Scientific Instruments PROM/RAM Board	95.00
MiniMicroMart C80-1702-I (all except PROMS)	49.95
MiniMicroMart C80-2708-2 (all except PROMS)	49.95
MiniMicroMart C80-256 (boot strap board, fuse link)	34.95
MITS PMC (2K)	85.00
Processor Technology 2KRO	65.00
Seals Electronics 4KROM	119.00
Solid State Music MB-3 2K (8 1702As)	105.00
Solid State Music MB-3 4K (16 1702As)	145.00
Solid State Music MB-3 (without PROMs)	65.00
Solid State Music MB-8 (2708)	85.00
Vector Graphic Reset and Go PROM/RAM	89.00
Xybek PRAMMER for 1702 (with a 1702 & RAM)	189.00

PIASTRE DI CONTROLLO MEMORIA

IMSAI IMM ROM Control Kit	299.00
IMSAI IMM EROM Control Kit	499.00

PIASTRE HARDWARE DI MOLTIPLICAZIONE/DIVISIONE

GNAT 8006 Module (5 u-sec. process time)	225.00
GNAT 8006 Module (2.5 u-sec. process time)	275.00
North Star Computers (floating point)	359.00

PIASTRE DI INTERFACCIA DI CALCOLATORE

COMPU/TIME CT 100	195.00
COMPU/TIME C 101	149.00
MiniMicroMart C80-SCI-300	99.95

PIASTRE DI SINTETIZZAZIONE VOCALE

Ai Cybernetic Systems Model 1000	325.00
Computalker Speech Synthesizer CT-I	395.00
Logistics Synthesizer (multipurpose)	525.00

PIASTRE DI RICONOSCIMENTO VOCALE

Heuristic Speechlab	245.00
Phonics SR/8 (assembled)	550.00

«KITS» DI INTERFACCIA A JOYSTICK

Cromemco Joystick Kit & D+7AI/O	210.00
Cromemco Dual Joystick Kits & D+7AI/O	275.00

PIASTRE DI INTERRUZIONE

Cromemco TU-ART	195.00
El Paso Computer Group (board only)	20.00
IMSAI PIC-8 (with internal clock)	125.00
MITS 88-VI/RTC	136.00

«REAL-TIME CLOCK»

Comptek CL2400	98.00
COMPU/TIME CT 100	195.00
COMPU/TIME T 102	165.00
International Data Systems SMP-88	96.00
Lincoln Semiconductor Clock and Display Driver	95.00

CONTROLLO ALIMENTAZIONE C.A

Comptek PC3216 Control Logic Interface	189.00
Comptek PC3216 & PC3202 Power Control Unit	228.50
Comptek PC3216 & 16 PC3202 16 Channel System	821.00
Comptek PC3232 Control Logic Interface	299.00
E.E. & P.S. 115V I/O	249.00
Müllen Relay/Opto Isolator Control Board	117.00

PIASTRE DI BLACK-UP DI BATTERIA

Seals Electronics BBUC (12 amper hours)	55.00
E.E. & P.S.	55.00

PIASTRE DI SINTETIZZAZIONE MUSICALE

ALF Quad Cromatic Pitch Generator (1 channel)	111.00
ALF Quad Cromatic Pitch Generator (2 channels)	127.00
ALF Quad Cromatic Pitch Generator (3 channels)	143.00
ALF Quad Cromatic Pitch Generator (4 channels)	159.00
Cybercam 581 Synthesizer Kit	250.00
Galazy Systems MG-1	299.00
Logistics Synthesizer (multipurpose)	525.00
SRS Polyphonic Synthesizer SRS-320 (assembled)	175.00
SRS Polyphonic Synthesizer SRS-321 for the SRS-320	175.00

PIASTRE DI INTERFACCIA DI STAMPANTE

Peripheral Vision PRT-KC Printer Kit	495.00

PIASTRE «CONTATORE DI FREQUENZA»

International Data Systems 88-UFC	149.00

PIASTRE INTERFACCIA IBM SELECTRIC

Micromation TYPEAWAY	225.00

PIASTRE INTERFACCIA PARALLELA

Advanced Microcomputer Products (3P+S compatible)	125.00
Cromemco D+7AIO (one part with seven analog parts)	145.00
Cromemco TU-ART (2 parts)	195.00
IMSAI PIO 4-I (one port without cables)	93.00
IMSAI PIO 4-I & PIOM (two ports without cables)	115.00
IMSAI PIO 4-I & PIOM (three ports without cables)	137.00
IMSAI PIO 4-4 (four ports without cables)	156.00
IMSAI PIO 6-3 (three ports and bus without cables)	139.00
IMSAI PIO 6-6 (six ports and bus without cables)	169.00
IMSAI MOI (two ports & serial & tape interface)	195.00
MicroLogic M712 (one port)	69.95
MiniMicroMart C80-P I/O (two ports)	49.95
MiniMicroMart C80-P I/O with cables C80-P I/O-540	57.45
MITS 88-4PIO (one port)	105.00
MITS 88-4PIO + PP (two ports)	148.00
MITS 88-4PIO + 2PP (three ports)	191.00
MITS 88-4PIO + 3PP (four ports)	234.00
Morrow Intelligent Cassette with one port	102.00
PolyMorphic VTI/32 (one input port with video)	185.00
PolyMorphic VTI/64 (one input port with video)	210.00

Processor Technology 3P+S (with sockets)	149.00
Solid State Music I/O-I (one port)	42.00
Solid State Music I/O-I (PC board only)	25.00
Solid State Music I/O-2 (two ports)	47.50
Solid State Music I/O-2 (PC board only)	25.00
Technical Design Labs Z Monitor Board (one port)	295.00
WIZARD PSIOB (3P+S compatible)	125.00

PIASTRE PROTOTIPO

Advanced Microcomputer Products Universal Proto	39.95
Artec GP-100	20.00
Cromemco WWB-2K	35.00
Electronic Control Technology PB-I	22.00
E.E. & P.S. Wire Wrap	39.00
E&L Instruments Breadboarding/Interfacing Station	241.50
Electronic Control Technology PB-I	28.00
Galaxy Systems PB-I	30.00
Harnestead Technology HTC-88P (QT sockets)	138.00
Harnestead Technology HTC-88PF (fell pattern)	38.00
IMSAI GP-88	39.80
IMSAI 88C-5 & P106-6 Intelligent Breadboard System	699.00
IMSAI 88C-3 & P106-3 Intelligent Breadboard System	464.00
MiniMicroMart C-80-WW (wire wrap type)	19.95
MiniMicroMart C-80-DIP (for point to point)	18.95
MiniMicroMart C-80-BUS-WW (wire wrap)	21.95
MiniMicroMart C-80-BUS-WW-125 (with components)	27.45
MiniMicroMart C-80-DIP-BUS (for point to point)	20.95
MiniMicroMart C-80-DIP-BUS-125 (with components)	26.45
MITS 88-PPCB	45.00
MITS 88-WWB	20.00
PolyMorphics Poly I/O	55.00
Processor Technology WWB	40.00
Sargent's Dist. Co.	25.00
Seals Electronics WWC	37.50
Tarbell Electronics	28.00
Vector 8800V	19.95
Vector 8800-A	29.95
Vector 8800-B	89.00

PIASTRE DI ESPANSIONE

Advanced Microcomputer Products Extender	34.95
Artec EXT-100	12.00
Cromemco EXC-2	35.00
E.E. & P.S. Extender W/C	34.00
Galaxy Systems EX-I	25.00
IMSAI EXT	39.00
MiniMicroMart C-80-EXC	24.95

Mullen (with logic probe)	35.00
Processor Technology EXB	35.00
Seals Electronics EXT	29.00
Solid State Music (less connectors)	8.00
Solid State Music (w/w connector)	12.50
Suntronics EXT-I	9.95
Vector 3690-12 (assembled)	25.00

PIASTRE ADATTAMENTO

MiniMicroMart C80-8A (for MOD 8/C-MOD 80 boards)	19.95
Forethought Products KIMSI (for KIM)	125.00

SCHEDE, GABBIA E/O PIASTRA MADRE

Advanced Microcomputer Products 8 slot MS w/connectors	79.95
Byte, Inc. Byt-8	229.00
Computer Data Systems Versatile CRT (assembled)	699.95
Electronic Control Technology ECT-100	100.00
Electronic Control Technology MB-20	60.00
Godbout Motherboard (10 slot)	85..00
Godbout Motherboard (18 slot)	118.00
Integrand Research Corp. 808	200.00
Integrand Research Corp. 808A	275.00
MiniMicroMart Expander (4 slots)	10.95
MiniMicroMart Expander (9 slots)	17.95
Morrow MotherBoard	76.00
Objective Design Crate Book (plans only)	19.95
PolyMorphic P+S Chassis	235.00
TEI Model MCS-112	316.00
T&H Engineering Low Cost Buses	149.00
Vector 18 Slot Motherboard	49.00

PIASTRE DI TERMINAZIONE

Godbout	25.00

PIASTRE INTERFACCIA VIDEO BIANCO E NERO

Computer Kits INTELLITERM (characters)	395.00
Computer Graphics GDT-I (graphics and light pen)	185.00
Environmental Interface II (monitor)	245.00
Environmental Interface III (oscilloscope)	495.00
Kent-Moore alpha (assembled)	107.00
Kent-Moore graphic (assembled)	137.00

Micro GRAPHICS "THE DEALER" (graphics and characters)	249.00
MiniMicroMart C80-VBA	149.95
MiniTerm Associates MERLIN (without memory)	269.00
MiniTerm Associates MERLIN (with memory)	303.95
MiniTerm Associates MERLIN Super Dense Graphics	308.00
Polymorphics VTI/64 (graphics and characters)	210.00
Processor Technology VDM-I (characters)	199.00
Solid State Music 64 x 16 (graphics and characters)	179.95

PIASTRE INTERFACCIA VIDEO-COLORE

Cromemco TV DAZZLER (graphics)	215.00

PIASTRE INTERFACCIA CAMERA TV

Cromemco 88-CCC-K	195.00
Cromemco 88-CCC-K with Camera Kit 88-ACC-K	390.00
Environmental Interface I	295.00
Environmental Interface with camera	595.00

Affordable Computer Products
Byte Shop No. 2
3400 El Camino Real
Santa Clara, CA 95051
(408) 249-4221

Advanced Microcomputer Products
P.O. Box 17329
Irvine, CA 92713
(714) 558-8813

Ai Cybernetic Systems
P.O. Box 4691
University Park, NM 88003

ALF Products, Inc.
128 S. Taft
Lakewood, CO 80228

Alpha Micro Systems
17875 N. SkyPark North
Irvine, CA 92714
(714) 957-1404

Altair (see MITS)

Artec Electronics, Inc.
605 Old Country Road
San Carlos, CA 94070
(415) 592-2740

Associated Electronics
12444 Lambert Circle
Garden Grove, CA 92641
(714) 539-0735

Base-2, Inc.
P.O. Box 9941
Marina del Rey, CA 90291

Byte Shop
1450 Koll Circle, No. 105
San Jose, CA 95112

CGRS Microtech, Inc.
Unknown

CHP, Inc.
P.O. Box 18113
San Jose, CA 95158

Comptek
P.O. Box 516
La Canada, CA 91011
(213) 790-7957

Computalker Consultants
P.O. Box 1951
Santa Monica, CA 90406

Computer Data Systems
English Village, Atram 3
Newark, DE 19711

Computer Kits Inc.
1044 University Avenue
Berkeley, CA 94710
(415) 845-5300

Computer Graphics Associates
56 Sicker Road
Latham, NY 12110

Computer Hobbyist Products, Inc.
P.O. Box 18113
San Jose, CA 95158
(408) 629-9108

COMPU/TIME
P.O. Box 417
Huntington Beach, CA 92648
(714) 638-2094

Computer Power & Light
12321 Ventura Blvd.
Studio City, CA 91604
(213) 760-0405

Crea Comp System, Inc.
Suite 305
4175 Veterans Highways
Ronkonkoma, NY 11779
(516) 585-1606

Crestline Micro Systems
P.O. Box 3313
Riverside, CA 92519

Cromemco
2432 Charleston Road
Mountain View, CA 94043
(415) 964-7400

Cybercom
2102A Walsh Avenue
Santa Clara, CA 95050
(408) 246-2707

DAJEN
David C. Jenkins
7214 Springleaf Court
Citrus Heights, CA 95610
(916) 723-1050

Data Sync
201 W. Mill
Santa Maria, CA 93454
(805) 963-8678

DigiComm
6205 Rose Court
Roseville, CA 95678

Digital Systems
1154 Dunsmuir Place
Livermore, CA
(415) 413-4078

Digiteck
P.O. Box 6838
Grosse Point, Michigan 48236

Duston, Forrest
885 Aster Avenue
Palatine, Il 60067

Dutronics
P.O. Box 9160
Stockton, CA 94608

E & L Instruments, Inc.
61 First Street
Derby, Conn. 06418
(203) 735-8774

E.E. & P.S.
Electronic Eng. & Production Service
Route No. 2
Louisville, Tennessee
(615) 984-9640

Electronic Control Technology
P.O. Box 6
Union City, NJ 07083

El Paso Computer Group
9716 Saigon Drive
El Pase, TX 79925

Environmental Interfaces
3207 Meadowbrook Blvd.
Cleveland, Ohio 44118
(216) 371-8482

Equinox Division
Parasitic Engineering
P.O. Box 6314
Albany, CA 94706
(800) 648-5311

Extensys Corp.
592 Weddell Drive, S-3
Sunnyvale, CA 94086
(408) 734-1525

Forethought Products
P.O. Box 386-A
Coburg, Oregon 97401

Franklin Electric Co.
733 Lakefield Road
Westlake Village, CA 91361
(805) 497-7755

Galaxy Systems
P.O. Box 2475
Woodland Hills, CA 91364
(213) 888-7233

GNAT Computers
8869 Balboa, Unit C
San Diego, CA 12123

Godbout Electronics
Box 2355
Oakland Airport, CA 94614

Hayes
P.O. Box 9884
Atlanta, GA 30319
(404) 231-0574

Heuristic, Inc.
900 N. San Antonio Road
Suite C-1
Los Altos, CA 94022

Hornestead Technologies Corp.
891 Briarcliff Road N.E.
Suite B-11
Atlanta, GA 30306

iCOM Division
6741 Variel Avenue
Conoga Park, CA 91303
(213) 348-1391

IBEX
1010 Morse Avenue, No. 5
Sunnyvale, CA 94086
739-3770

I M S Associates, Inc.
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093

INFO 2000
P.O. Box 316
Culver City, CA 90230

Integrand Research Corp.
8474 Avenue 296
Visalia, CA 93277
(209) 733-9288

International Data Systems
400 North Washington Street,
Suite 200
Falls Church, VA 22046
(703) 536-7373

Kent-Moore Instrument Co.
P.O. Box 507
Industrial Avenue
Pioneer, Ohio 43554
(419) 737-2352

Lewis and Associates
68 Post Street, Suite 506
San Francisco, CA 94104
(415) 391-1498

Lincoln Semiconductor
P.O. Box 68
Milpitas, CA 95035
(408) 734-8020

Logistics
Box 9970
Marina Del Rey, CA 90291

North Star Computers
2465 Fourth Street
Berkeley, CA 94710

MECA
7344 Warnego Trail
Yucca Valley, CA 92284
(714) 365-7686

Micro Data
3199 Trinity Place
San Jose, CA 95124

Microdesign
8187 Havasu Circle
Buena Park, CA 90621
(415) 465-1861

Micro Designs, Inc.
499 Embarcadero
Oakland, CA 94606
(415) 465-1861

MicroGRAPHICS
P.O. Box 2189, Station A
Champaign, IL 61820

MicroLogic
P.O. Box 55484
Indianapolis, IN 46220

Micromation
524 Union Street
San Francisco, CA 94133
(415) 398-0289

Micronics, Inc.
P.O. Box 3514
Greenville, NC 27834

Micropolis Corp.
9017 Reseda Blvd.
Northridge, CA 91324

Midwest Scientific Instruments
220 West Cedar
Olathe, Kansas 66061

MIKRA-D, Inc.
P.O. Box 403
Hollister, Mass. 01746

Mini Micro Mart
1618 James Street
Syrecuse, NY 13203

MiniTerm Associates
Box 268
Bedford, Mass. 01730

MITS (Altair)
2450 Alamo S. E.
Albuquerque, NM 87106

Morrow's Micro-Stuff
Box 6194
Albany, CA 94706

MRS
P.O. Box 1220
Hawthorne, CA 90250

Mullen Computer Boards
Box 6214
Hayward, CA 94545

Mountain Hardware
Box 1133
Ben Lamand, CA 95005

National Multiplex Corp.
3474 Rand Avenue, Box 288
South Plainfield, NJ 07080

Objective Design, Inc.
P.O. Box 7536 Univ. Station
Provo, Utah 84602

PerCom Data Company
4021 Windsor
Garland, TX 75042

Peripheral Vision
P.O. Box 6267
Denver, Colorado 80206

Phonics, Inc.
P.O. Box 62275
Sunnyvale, CA 94086

Polymorphic Systems
737 S. Kellogg
Galeta, CA 94608

Prime Rodix Inc.
P.O. Box 11245
Denver, Colorado 80211

Processor Applications, Ltd.
2801 East Valley Veiw Avenue
West Covina, CA 91792

Processor Technology
6200-L Hollis Street
Emeryville, CA 94608

Quay Corporation
P.O. Box 386
Freehold, NJ 07728

Realistic Controls Corporation
3530 Warrensville Center Road
Cleveland, Ohio 44122

R.H.S. Marketing
2233 El Camino Real
Palo Alto, CA 94306

RO-CHE Systems
7101 Mammoth Avenue
Van Nuys, CA 91405

S. D. Sales
P.O. Box 28810
Dallas, Texas 75228

Sargent's Dist. Co.
4209 Knoxville
Lakewood, CA 90713

Scientific Research Instruments
P.O. Drawer C
Marcy, NJ 13403

Seals Electronics
Box 11651
Knoxville, TN 37919

Smoke Signal Boardcasting
P.O. Box 2017
Hollywood, CA 90028

Solid State Music
MIKOS
419 Portofino Drive
San Carlos, CA 94070

Stillman Research Systems (SRS)
P.O. Box 14036
Phoenix, AZ 85063

Suntronics Company
360 Merrimack Street
Lawrence, MA 01843

Synetic Designs Company
P.O. Box 2627
Pomona, CA 91766

Szerlip Enterprises
1414 W. 259th Street
Harbor City, CA 90710

TEI Inc.
7231 Fondren Road
Houston, Texas 77036

T&H Engineering
P.O. Box 352
Cardiff, CA 92007

Tarbell Electronics
20620 South Leapwood Avenue
Suite P
Carson, CA 90746

Technical Design Labs Inc.
342 Columbus Avenue
Trenton, NJ 08629

Vandenberg Data Products
P.O. Box 2507
Santa Maria, CA 93454

Vector Electronics Company, Inc.
12460 Gladstone Avenue
Sylmar, CA 91342

Vector Graphic Inc.
717 Lakefield Road, Suite F
Westlake Village, CA 91361

Western Data Systems
3650 Charles Street, No. Z
Santa Clara, CA 95050

WIZARD Engineering
8205 Ronson Road, Suite C
San Diego, CA 92111

Xybek
P.O. Box 4925
Stanford, CA 94305

APPENDICE C

TABELLA DI CONVERSIONE

DECIMALE	BINARIO	ESADECIMALE	OTTALE
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

APPENDICE D

SEGNALI RS232C

PIN	FUNZIONI
1	Massa di protezione
2	Trasmissione dati all'apparecchiatura di comunicazione (T x D)
3	Ricezione dati dall'apparecchiatura di comunicazione (R x D)
4	Richiesta trasmissione all'apparecchiatura di comunicazione (RTS)
5	Pronto per la trasmissione dall'apparecchiatura di comunicazione (CTS)
6	«Data set» pronto dall'apparecchiatura di comunicazione (DSR)
7	Ritorno corrente
8	Rivelazione portante dall'apparecchiatura di comunicazione (DCD)
20	Terminale pronto all'apparecchiatura di comunicazione (DTR)

APPENDICE E

SEGNALI IEEE-488

D101-D108	Vie dati	Trasporto dati
DAV	Validità dati	Indica se le linee dati contengono dati stabili
NRFD	Non pronto per dati	Diventa falso quando tutti i dispositivi hanno accettato dati
NDAC	Dati non accettati	Diventa falso quando tutti i dispositivi hanno accettato dati
ATN	Attenzione	Indica se le linee dati trasferiscono indirizzi oppure dati
IFC	Ripristino di interfaccia	È un segnale di reset
SRQ	Richiesta	Segnale di interruzione
REN	Servizio abilitazione remota	Seleziona il funzionamento del pannello frontale
EOI	Fine della identificazione	Fine del trasporto o della frequenza di registrazione

APPENDICE F

ACRONIMI

AC — Alternating Current
Corrente alternata

ACC — Accumulator
Accumulatore

ACK — Acknowledge
Conferma ricezione

A/D — Analog to Digital
Da analogico a digitale

ADCCP — Advanced Data Communication Control Procedure
Procedura di controllo di comunicazione dati avanzata

ALU — Arithmetic-Logic Unit
Unità aritmetico-logica

ANSI — American National Standards Institute
American National Standard Institute

ASCII — American Standard Code for Information Interchange
American Standard Code for Information Interchange

ASR — Automatic Send and Receive
Trasmissione-ricezione automatiche

BCD — Binary-Coded-Decimal
Decimale codificato in binario

BCR — Byte Count Register
Registro conteggio di byte

BPS — Bits Per Second
Bit per secondo

BRA — Branch, go to
Ramificazione (branch), «go to»

BSC — Binary Synchronous Communication
Comunicazione sincrona binaria

C — Carry
Riporto

CAD — Computer-Aided-Design
Progetto con l'ausilio del calcolatore

CAM — Contents-Addressable Memory
Memoria indirizzabile su contenuto

CCD — Charge-Coupled Device
Dispositivo ad accoppiamento di carica

CE — Chip Enable
Abilitazione di modulo

CLK — Clock
«Clock» — Temporizzatore

CML — Current Mode Logic
Logica di modo corrente

CMOS — Complementary MOS
MOS complementare

CPG — Clock Pulse Generator
Generatore di impulsi di clock

CPS — Characters Per Second
Caratteri al secondo

CPU Central Processor Unit
Unità Centrale di elaborazione

CR Card Reader; Carriage Return
Lettore di Carta – Ritorno carrello

CRC Cyclic Redundancy Check
Controllo di ridondanza ciclica

CROM Control-ROM
ROM di controllo

CRT Cathode Ray Tube
Tubo a raggi catodici

CRTC CRT Controller
Controllore di CRT

CS Chip Select
Selettore di modulo

CTS Clear to Send
«Clear to send»

CU Control Unit
Unità di controllo

CY Carry
Riporto

D Data
Dati

D/A Digital to Analog
Da digitale ad analogico

DC Direct Current
Corrente continua

DC Don't Care
Qualsiasi valore («don't care»)

DCD Data Carrier Detect
Rivelatore di portante dati

DIP Dual In-Line Package
Incapsulamento «Dual In-Line»

DMA Direct Memory Access
Accesso diretto in memoria

DMAC DMA Controller
Controllore DMA

DMOS Double-Diffused MOS
MOS a doppia diffusione

DNC Direct Numerical Control
Controllo numerico diretto

DOS Disk Operating System
Sistema operativo basato su disco

DPM Digital Panel Meter
Misura su pannello digitale

DTL Diode-Transistor Logic
Logica basata su diodi-transistori

DTR Data Terminal Ready
«Data terminal ready»

D0-7 Data Lines 0 Through 7
Vie dati da 0 a 7

E Empty; Enable (Clock)
Vuoto; Abilitazione (Clock)

EAROM Electrically Alterable ROM
ROM alterabile elettricamente

EBCDIC Extended Binary - Coded-Decimal Information Code
Codice d'informazione digitale codificato in binario esteso

ECL Emitter Coupled Logic
Logica di accoppiamento di emettitore

EDP Electronic Data Processing
Elaborazione dati elettronica

EFL Emitter Follower Logic
Logica ad inseguimento di emettitore

EMI Electro Magnetic Interference
Interfaccia elettromagnetica

EOC End of Conversion
Fine della conversione

EOF End of File
Fine del file

EOR Exclusive OR
OR esclusivo

EOT End of Text, Tape
Fine della trasmissione

EPROM Erasable PROM
PROM cancellabile

FAMOS Floating-Gate Avalanche MOS
Porta fluttuante a valanga MOS

FDC Floppy-Disk Controller
Controllore di disco floppy

FDM Frequency-Division Multiplexing
Multiplazione a divisione di frequenza

FET Field-Effect Transistor
Transistore ad effetto di campo

FF Flip-Flop
Flip-flop

FIFO First-In-First-Out
Primo in ingresso-primo in uscita

FPLA Field PLA
PLA mediante campo

FSK Frequency-Shift-Keying
Chiave a scorrimento di frequenza

G (carry) Generate
Generazione (di carry)

GP General-Purpose
Per applicazioni generali

GPIB General-Purpose Interface Bus
Bus di interfaccia GP

HDLC High Level Data Link Control
Accoppiamento per dati su livello alto

HEX Exadecimal
Esadecimale

HPIB Hewlett-Packard Interface Bus
Bus di interfaccia Hewlett-Packard

I Interrupt/Interrupt Mask
Interruzione/maschera di interruzione

IC Integrated Circuit = Chip
Circuito integrato (Chip)

INT Interrupt
Interruzione

I/O Input/Output
Ingresso/uscita

IOCS I/O Control System
Sistema di controllo I/O

IRQ Interrupt Request
Richiesta di interruzione

I^2L Integrated Injection Logic
Logica di integrazione ad iniezione di corrente

JAN Joint Army-Navy
Joint Army-Navy

JP Jump
Salto

K (1024) Kilo
Kilo (1024)

KSR Keyboard-Send-Receive
Tastiera di trasmissione/ricezione

LCD Liquid-Crystal Display
Visualizzazione a cristalli liquidi

LED Light Emitting Diode
Diodo ad emissione di luce

LIFO Last-In-First-Out
Ultimo entrato-primo in uscita

LOC Loop On-Line Control
Controllo di anello (loop) «on line»

LP Line Printer
Stampante per linee

LPM Lines Per Minute
Linee al minuto

LPS Low-Power Shottky
Low-Power Shottky

LRC Longitudinal Redundancy Check
Controllo a ridondanza longitudinale

LSB Least Significant Bit
Bit meno significativo

LSI Large Scale Integration
Integrazione a larga scala

MNOS Metal Nitride Oxide Semicon-

ductor
Metallo-Nitruro-Ossido-Semiconduttore

MOS — Metal Oxide Semiconductor
Metallo-Ossido-Semiconduttore

MPU — Microprocessor Unit
Unità microprocessore

MSB — Most Significant Bit
Bit più significativo

MSI — Medium Scale Integration
Integrazione a media scala

MTBF — Mean Time Between Failures
Intervallo medio tra guasti successivi

MUX — Multiplexer
Multiplatore

N — Negative (Sign Bit)
(bit di segno) negativo

NDRO — Non-Destructive Read-Out
Lettura non distruttiva

NMOS — N-Channel MOS
MOS a canale N

NVM — Non-Volatile Memory
Memoria non volatile

OCR — Optical Character Reader
Lettore ottico di caratteri

OEM — Original Equipment Manufacturer
Costruttore di apparecchiatura originale

OP — Operation
Operazione

OV — Overflow
Overflow

P — Parity; (carry) Propagate
Parità; propagazione (del carry)

PABX — Private Automatic Branch Exchange
Commutatore automatico privato di agenzia

PBX — Private Branch Exchange
Commutatore privato di agenzia

PC — Printed Circuit; Program Counter
Circuito stampato; Contatore di programma

PCI/O — Program Controlled I/O
I/O controllato da programma

PCM — Pulse Code Mod.
Modulazione a codifica di impulso

PFR — Power-Fail Restart
Ripristino per caduta di alimentazione

PIC — Priority Interrupt Control
Controllo prioritario di interruzioni

PIO — Programmable I/O Chip/Interface
Circuito di interfaccia programmabile di I/O

PIT — Programmable Interval-Timer
Temporizzatore programmabile di intervallo

PLA — Programmable Logic-Array
Trama logica programmabile

PLL — Phase-Locked Loop
Anello ad aggancio di fase

PMOS — P-Channel MOS
MOS a canale P

POS — Point-of-Sale Terminal
Terminale per punto di vendita

PROM — (Field) Programmable ROM
ROM programmabile in «campo»

PSW — Program Status Word
Parola di stato di programma

PTP — Paper Tape Punch
Nastro di carta perforato

PTR — Paper Tape Reader
Lettore di nastro di carta

Q — AC extension
Estensione di AC

QPL Qualified Products List
Lista di prodotti omologati

R Read
Lettura

RALU Register Arithmetic Logic Unit
Registro di unità logico-aritmetica

RAM Random-Access-Memory
Memoria ad accesso casuale

RDSR Receiver Data Service Request
Richiesta di servizio dal ricevitore dati

RDY Ready
Pronto

RES Reset
«reset»

RF Radio Frequent
Radiofrequenza

RMS Root Mean Square
Valore quadratico medio

ROM Read-Only Memory
Memoria a sola lettura

RPROM Reprogrammable PROM
PROM programmabile

RPT Repeat
Ripeti

RS Register Select
Selezione di registro

RST Restart
Ri-inizializzazione

RTC Real-Time Clock
Clock «real-time»

RTS Request-To-Send
«Request-to-send»

R/W Read/Write Memory
Memoria di lettura/scrittura

Rx Receiver
Ricevitore

SAR Successive Approximation Register
Registro per approssimazioni successive

SDLC Synchronous Data Link Control
Controllo di giunzione dati sincrona

SEC Scanning Electron Microscope
Microscopio a scansione elettronica

SEM Standard Electronic Module
Modulo di elettronica standard

S/H Sample and Hold
Campionamento e mantenimento

S/N Signal to Noise
Segnale rispetto al rumore

SOS Silicon-On-Sapphire
Silicio su zaffiro

SR Service Request
Richiesta servizio

SSI Small Scale Integration
Integrazione su piccola scala

STB Strobe
Finestra di abilitazione (strobe)

SUB Subroutine
Subroutine (sottoprogramma)

TDM Time-Division Multiplexing
Multiplazione a divisione di tempo

TDSR Transmitter Data Service Request
Richiesta di servizio dati da trasmettitore

TSS Time-Sharing System
Sistema a divisione di tempo

TTL Transistor Transistor Logic
Logica basata su Transistori-transistori

TTY Teletypewriter
Telescrivente

Tx Transmitter
Trasmettitore

UART Universal Asynchronous Re-

	ceiver Transmitter *Trasmettitore-ricevitore universale asincrono*
μC	Microcomputer *Microcalcolatore*
μP	Microprocessor *Microprocessore*
USRT	Universal Synchronous Receiver Transmitter *Trasmettitore-ricevitore universale sincrono*
U-V	Ultra-Violet *Ultravioletto*
VMOS	Vertical MOS *MOS verticale*
V_{SS}	Ground *Massa*
W	Write *Scrittura*
WPM	Words Per Minute *Parole al minuto*
X	Index *Indice*
XOR	Exclusive OR *OR esclusivo*
Z	Zero Bit *Bit zero*
Φ	(Clock) Phase *Fase (di clock)*

Edizione Italiana del
MICROPROCESSOR INTERFACING TECNIQUES

della SYBEX Inc.

JACKSON
ITALIANA
EDITRICE

AUSTIN LESEA
RODNAY ZAKS

Tecniche d'interfacciamento dei microprocessori

25